

論文内容の要旨

博士論文題目 オブジェクト指向に基づく並列化コンパイラフレームワークの研究
氏 名 大森 洋一

本論文は、従来の逐次型コンパイラよりも複雑な内部構造をもつ並列化コンパイラ的设计論へのオブジェクト指向设计論の適用に関する研究をまとめたものであり、序論・結論を含め六つの章からなっている。

第一章では、本研究の背景と目的、関連研究、並列化コンパイラ設計の課題を述べるとともに、本論文内容を概説している。

第二章では、従来のコンパイラの内部データフローを分析し、当該コンパイラを並列化コンパイラへと機能拡張する際、並列化フェーズの位置、拡張性の高い共有データ的设计に関する問題点を明らかにし、これらの問題に対するオブジェクト指向设计論の適用の意義を述べている。

第三章では、オブジェクト指向设计論であるOMT (Object Modeling Technique) を並列化コンパイラ設計に適用している。OMTによる並列化コンパイラ設計の問題分析に際して、ストリームという概念を導入し、並列化コンパイラの機能を、フロントエンド、バックエンド、および中間表現の構成に三分割して詳細化する方針を提唱している。これにより、従来のコンパイラからの変更点が小さいフロントエンドに関しては従来と同様の効率的な設計が、バックエンドに関してはテンプレートを用いたクラスの階層化が容易になる。

第四章では、OMTを利用して並列性の明示的な表現が可能な、クラス化されたコンパイラ中間表現を設計している。具体的には、階層化タスクグラフ (Hierarchical Task Graph: HTG) と RTL (Register Transfer Language) の二階層からなる中間表現を用いて、基本ブロックレベルと命令レベルの並列性の表現を可能としている。また、任意のレベルにおける並列表現の基底クラスとして利用するために、並列性を表現する抽象クラスを抽出している。さらに、本中間表現にループに関するクラスを追加し、並列化研究の中核をなすループ並列化フェーズをOMTを用いて詳細設計している。その結果、本フェーズは、ループ中間表現、並列化アルゴリズム記述、および出力ハードウェア情報の三つのモジュールから構成されることを示している。また、並列性抽出の中核となる並列化アルゴリズム記述モジュールに関しては、クラス化により、類似のアルゴリズム間の変更が容易になることを、分離テストとバナジーテストの二つの例を用いて実証している。最後に、本設計法に基づき、ループ並列化フェーズに相当する部分のプロトタイプを約五千行からなるC++で記述している。さらに、本プロトタイプを同じくC++で記述された並列化コンパイラ SUIF (スタンフォード大学) の同一機能部分と手続きやクラスのサイズなどに関して比較・検討している。その結果、本プロトタイプはSUIFよりもクラスの独立性など、設計上の改善がなされていることを示している。

第五章では、本中間表現のバックエンドへの応用について述べている。具体的には、出力ハードウェア情報を保持するクラスの詳細化と、ループ中間表現の拡張および整理を行なっている。また、複数の並列化アルゴリズム間での共有データの一貫性保持にコピーアンドコミット方式を提案している。

第六章では、以上の研究成果についてまとめ、今後の課題を明らかにしている。

論文審査結果の要旨

本論文は、オブジェクト指向設計論の一つであるOMT (Object Modeling Technique) を並列化コンパイラの設計に適用し、並列化コンパイラの機能をフロントエンド、バックエンド、および中間表現に三分割して詳細化するための概念であるストリームの提案、複数レベルの並列性の表現を可能とするクラス化された中間表現の設計、さらに、本中間表現を利用してループ並列化フェーズの詳細設計とプロトタイプを構築するとともに本プロトタイプの評価により、本構成法の有効性を示したものである。本論文の主な成果は以下に要約される。

1. 並列化コンパイラを設計する際のオブジェクト分割の方針として、従来のコンパイラからの変更点が小さいフロントエンドと、複雑な並列化アルゴリズムへの対応が求められるバックエンドを一貫して扱えるように、データフローに基づいて、並列化コンパイラの機能をフロントエンド、バックエンド、および中間表現の三分割により詳細化するための概念であるストリームを提案した。
2. HTG (Hierarchical Task Graph) および RTL (Register Transfer Language) の二つの階層により、これらのレベルの並列性を表現できる中間表現をクラスを用いて設計した。さらに、これら複数のレベルに対応する並列性を表現するクラスの基底となる抽象クラスを抽出し、これによって各レベルの並列性表現に統一した見方を与えた。
3. 並列化研究の中核をなすループ並列化フェーズをOMTを用いて詳細設計し、本フェーズが、ループ中間表現、並列化アルゴリズム記述、および出力ハードウェア情報の三つのモジュールから構成されることを示した。また、並列性抽出の中核となる並列化アルゴリズム記述モジュールに関しては、クラス化により、類似のアルゴリズム間の変更が容易になることを、分離テストとバナジーテストの二つの例を用いて実証した。
4. 上記3. の詳細設計に基づき、ループ並列化フェーズに相当する部分のプロトタイプを約五千行からなるC++で実際に構築した。さらに、本プロトタイプを同じくC++で記述された並列化コンパイラ SUIF (スタンフォード大学) の同一機能部分と手続きやクラスのサイズなどに関して比較・検討した。その結果、本プロトタイプはSUIFよりもクラスの独立性など、設計上の改善がなされていることを実証した。

以上のように、本論文は、逐次型コンパイラ設計を踏襲して設計され、柔軟性に欠けていた並列化コンパイラの設計法に、オブジェクト指向設計論を適用することの有効性を示唆したものであり、並列化コンパイラの設計法の分野において、学術上、実際上寄与するところが少なくない。よって、本論文は、博士(工学)の学位論文として価値あるものと認める。