

| | |
|----------|---|
| 氏名 | ひら いし たすく 平 石 拓 |
| 学位(専攻分野) | 博 士 (情 報 学) |
| 学位記番号 | 情 博 第 285 号 |
| 学位授与の日付 | 平 成 20 年 1 月 23 日 |
| 学位授与の要件 | 学 位 規 則 第 4 条 第 1 項 該 当 |
| 研究科・専攻 | 情 報 学 研 究 科 通 信 情 報 シ ス テ ム 専 攻 |
| 学位論文題目 | Transformation-based Implementation of S-expression Based C Languages (S式ベースC言語およびその拡張言語の変形に基づく実装) |
| 論文調査委員 | (主 査) 教 授 湯 淺 太 一 教 授 佐 藤 雅 彦 教 授 奥 乃 博 |

論 文 内 容 の 要 旨

プログラミング言語の開発者が高水準言語を開発する際、アセンブリ言語ではなくC言語への変換器としてコンパイラを実装することがある。そうすることで、最適化などの複雑な処理を既存のCコンパイラに任せられ実装が容易になる。さらに、プラットフォームごとにコンパイラを実装する必要がなくなるため、実装コストを削減することができる。

本論文では、このようなC言語への変換による高水準言語の開発を支援するためのフレームワークとして「SC言語処理系」を提案している。中間言語としてC言語を利用する際の問題として

- ・ C言語の構文は直接の解析・変形に適さないため、構文解析を前処理として行う必要があり、その実装が開発の負担になること。
- ・ 標準のC言語は、実行スタックにアクセスする能力を持たないため、ごみ集めなどの機能をC言語への変換で実装することが困難なこと。

という二点を特に取り上げ、解決策を示している。また実際に、Tascellという動的負荷分散による並列処理言語を、本論文で提案している手法を用いて実装している。

本論文の構成は以下の通りである。

第1章 Introduction

研究の背景、目的について述べ、貢献を列挙したあと、論文の構成を説明している。

第2章 The SC Language System

本論文の主題であるSC言語処理系について述べている。SC-1言語というS式ベースの(Lisp言語風の構文を持つ)C言語を提供することにより、S式ベースの高水準言語のコンパイラを、SC-1言語をターゲットとするS式の変形規則を記述するだけで実装できるようにしている。さらに、S式パターンマッチングなどの機能を備えた拡張Lisp言語を提供することで、変形規則の記述を容易にしている。

第3章 Evaluation and Discussion

第2章で提案したSC言語処理系による言語実装手法について、実装コストの定量的評価や、変形規則の記述性についての議論を行っている。

第4章 Transformation-based Implementation of Lightweight Nested Functions

第2章で提案した手法を実際に用いて実装した言語、LW-SC言語を、SC言語処理系による高水準言語の実装の具体例として紹介している。また、LW-SC言語が提供する言語機能である低コストの「入れ子関数」(関数の中で定義できる関数)により、C言語がサポートしていない実行スタックへのアクセスが可能になり、ごみ集めなどの言語機能も標準のC言語への変換で実装できることを示している。

第5章 Using Existing C Header Files in SC Based on Translation from C to SC

C言語からSC-1言語の変換器を実装することにより、C言語で記述されたヘッダファイルをSC言語から利用できるようにしたことを述べている。その中の議論で、C言語のマクロからSC言語のマクロへの自動変換が不可能であると述べ、その問題に対する現実的な解決策を提案している。

第6章 Backtracking-based Load Balancing

Tascellという動的負荷分散による並列計算用言語について述べている。高性能な動的負荷分散を実現するためのアイデアとして、バックトラックを用いる手法を提案しており、既存の並列計算言語であるCilkと性能の比較を行い、高い性能を発揮することを示している。

実装には第2章で提案しているSC言語処理系と、第4章で提案している入れ子関数を利用したスタックアクセスの実装手法を利用することで、これらの手法が実際に有用であることを示している。

第7章 Related Work

本論文が提案する言語拡張の手法などについて、既存の関連研究をいくつか紹介している。

第8章 Conclusion and Future Work

本論文で得られた主要な結論についてまとめ、今後の展望について述べている。

論文審査の結果の要旨

C言語を高水準言語の実装用中間言語として利用することで、小さい開発コストでコンパイラの実装を行うことができる。しかし、C言語の能力は実際に様々な言語機能を実装するには不十分な場合がある。またコンパイラの実装の際、C言語風の構文で書かれたプログラムはいったん抽象構文木に変換する必要があるが、この変換器の実装コストも無視できない。本論文はこれらの問題を解決する手法を提案し、またその手法の複数の応用例を示したものである。

本論文の主要な貢献は以下のとおりである。

- ・ SC-1言語というS式ベースC言語を開発し、高水準言語のコンパイラ実装を、SC-1言語をターゲットとするS式の変形規則の記述のみによって行うことを提案した。また、その変形規則を記述、再利用するのに便利なパターンマッチングなどの拡張機能も提供することで、言語開発の効率良いプロトタイピングを可能にした。
- ・ C言語に入れ子関数の機能を追加したLW-SC言語というプログラミング言語を開発した。この言語を中間言語と利用することで、従来は困難であったごみ集めなどの言語機能のC言語への変換による実装を、統一的な手法で行うことを可能にした。またそのような入れ子関数を、実行性能を犠牲にしない形で実装する手法を提案した。
- ・ 動的負荷分散による並列計算の手法として、タスク分割時にバックトラックを行うという新しい手法を提案し、さらにそれを実現するためのプログラミング言語Tascellを開発した。既存の並列計算言語であるCilkとの性能比較も行い、多くのアプリケーションでより良い性能を発揮することを示した。
- ・ LW-SC言語やTascell言語の実装には本論文中で提案しているSC言語を用いる手法を、さらにTascell言語の実装ではLW-SC言語の入れ子関数を用いる手法も実際に利用しており、それぞれの手法の有用性を裏付けている。

以上要するに本論文は、今後プログラミング言語の開発者・研究者が様々な高水準言語の開発を行っていくためで有用であり、学術上、實際上寄与するところが少なくない。よって、本論文は博士（情報学）の学位論文として価値あるものと認める。また、平成19年12月17日実施した論文内容とそれに関連した試問の結果合格と認めた。