審 査 の 結 果 の 要 旨

氏　名　海野　広志

プログラムの形式的検証のために様々な手法が提案されているが、型に基づく解析は、プログラムの仕様を型として表現し、型推論によってプログラムが仕様を満たしているかどうかを自動的に検査する方法である。特に、値に依存できるように拡張された型である「依存型」を用いることにより、プログラムのより詳細な仕様を型として表現することが可能となる。しかし、依存型を含む型システムにおいては、型付け可能性は一般に決定不能であり、完全な型推論アルゴリズムは存在しない。

本論文では、高階関数や再帰的データ構造の表現可能な関数型言語に対して、依存型を含む型システムが導入され、その型システムのための二つの型推論アルゴリズムが提案され実装されている。

本論文の第1章では、以上で述べたような研究の背景と本論文の概要について述べられている。

第2章では、本論文の型システムによる解析の対象となる関数型言語が、その構文と操作的意味論とともに、説明されている。

第3章では、本論文の主題である型システムが詳しく述べられている。特に、前章の関数型言語の操作的意味論に対する型システムの健全性が証明されている。

第4章では、型推論のために、関数型言語のプログラムから、型に関する制約を生成する方法について述べられている。一般に、この制約を解くことにより、型推論が行われる。

以上の枠組みのもとで、第5章では、型推論の第一のアルゴリズムが述べられている。このアルゴリズムは、関数の呼び出しサイトから関数の仕様を推論し、それを後ろ向きに伝搬することによって、要求された仕様を検証するのに十分な程度に詳細な依存型を自動的に求めることができる。第5章では、このアルゴリズムのプロトタイプ実装および実際のプログラムを対象とした二つの実験についても報告されている。一番目の実験では、挿入ソートとマージソートを実装したプログラムが必ず整列されたリストを返すことを自動的に検証することに成功している。二番目の実験では、OCamlプログラミング言語のリスト操作ライブラリ関数群の依存型を、そのライブラリを呼び出している実際のOCamlプログラムを用いて推論した結果が述べられている。

第6章では、型推論の第二のアルゴリズムが述べられている。このアルゴリズムは、補間定理証明器を用いて、型推論が成功するか反例が見つかるまで徐々に依存型を詳細化する。特に、反例を出力することは、第一のアルゴリズムにはない優位点である。また、補間定理証明器を用いた詳細化を反復することにより、第一のアルゴリズムでは求められなかった型を推論することができる。第6章では、このアルゴリズムのプロトタイプ実装および実際のプログラムを対象とした二つの実験についても報告されている。特に、一番目の実験では、配列を操作するプログラムが配列境界エラーを起こさないことを、ユーザに依存型宣言を要求することなく検証できることが示されている。

第7章では関連研究、第8章では本論文の結論が述べられている。

以上をまとめると、本論文は、依存型を含む型システムに対して、二つの新たな推論アルゴリズムを開発することにより、実践的なプログラミング言語における依存型の応用可能性を、大きく広げたということができる。そして、この結果は、プログラムの形式的検証技術に対する十分な貢献と考えられる。

よって本論文は博士（情報理工学）の学位請求論文として合格と認められる。