論文要旨

学位論文題目：Abstracting Control with Dependent Types

（依存型付き言語への限定継続命令の導入）

氏名：叢 悠悠

プログラミングにおいて、「型」はデータの種類を表現するための概念であり、これを用いると、多くの実行時エラーを回避することができる。なかでも、「依存型」とよばれる種類の型は、数字や文字列のような値への依存を許すことで高い表現力を実現しており、意図しない関数の使用を防いだり、プログラムの満たす性質を証明したりする目的で用いられる。

一方、「制御演算子」は残りの計算を明示的に扱うためのツールであり、これを使うと、プログラムの流れを自由に操作することができる。特に、「限定継続命令」とよばれる種類の命令は、通常の関数のように振る舞う継続を捕捉することを特徴としており、プログラムの最適化や Web サーバの実装など、さまざまな場面で応用されている。

以上の観点から、依存型と限定継続命令を同時に扱うことができれば、信頼性の高いプログラムを簡潔に実装することが可能になると考えられる。しかし、これらの要素を安易に組み合わせると、型が静的に定まらない、あるいは、論理的に矛盾した体系が得られるということが知られている。こうした問題を解決するために、先行研究では、「型の中に現れてよいのは、複雑な制御を伴わない純粋な項のみ」という制限を設けながら、依存型と制御演算子を共存させる、というアプローチがとられている。

本研究は、安全なプログラムの効率的な実装を目標とし、Danvy and Filinski の shift/reset 演算子を依存型付き言語に導入する。以下、論文の構成と具体的な貢献を示す。

まず、shift と reset を依存型付き言語で扱う際に、どのような問題が起きるかを明らかにする。これらの演算子は、継続の合成性と、コンテキストが返す型の変更を許すことで、柔軟な制御を可能にしているが、依存型付き言語に導入した場合、この柔軟性によって、型判断に自由変数を含む型が現れてしまうことがある。そこで、型に現れる項への制限に加え、「依存型をもつ継続」と「継続への依存性」に対する制約を設ける。これによって、正しく型付けされた項が情報量のある型をもち、閉じた型のみからなる型判断をもつことを保証する。

必要な制限が分かったところで、実際に shift/reset をもつ小さな依存型付き言語 Dellina- を設計する。具体的なアイディアとしては、項の実行に伴うエフェクトの情報をもつ型判断を用いて、型の依存性に関する 3 つの制限を型規則の中に埋め込む。こうして得られた型システムは健全であること、つまり、正しく型が付いたプログラムは、実行時に型の不一致によるエラーが発生しないということが保証される。

次に、Dellina- に対する継続渡し形式への変換（CPS 変換）を与える。CPS 変換とは、直接形式で書かれたプログラムの継続を明示的にするためのプログラム変換であり、Dellina- においては、プログラム中の shift/reset を除去するという役割をもつ。依存型付き言語に対する CPS 変換は、長年不可能とされてきたが、最新の研究により、型の依存性を純粋な項に制限しつつ、非叙述的な多相を許せば、型を保存するような変換が与えられることが明らかになっている。本研究では、制御を伴う項のみを CPS に変換する選択的なアルゴリズムを採用することで、非叙述性を仮定せずに、型の保存性を保証する。これにより、叙述的な多相のみを許す Agda のような言語においても、CPS 変換の実装が可能となる。

Dellina- は表現力が限られた言語であるが、提案手法はより実用的な言語にも適用できる。これを示すために、Dellina- を多相型、universe の階層、および帰納的データ型で拡張し、言語 Dellina を得る。また、拡張した言語を使って、効率的なエラー処理をサポートする型安全なインタプリタを実装し、shift/reset と依存型を組み合わせることの有用性を示す。

制御演算子をもつ多くの言語と同様に、Dellina のプログラムは値呼びの意味論に基づいて実行される。一方で、依存型付き言語の多くは、名前呼びの意味論をもつ。そこで、名前呼びの Dellina- を設計し、値呼びの言語との違いを観察する。そのなかで、名前呼びの意味論のもとでは、「値」と「計算」の区別がより重要になることを示す。

最後に、shift/reset を用いて証明を行うということは何を意味するのか、また、Dellina ではどのような定理が証明できるのかを考察する。純粋なラムダ計算において、プログラムは証明、型は命題と理解されるが、Dellina では、純粋な項と型のみを証明あるいは命題としてみなす、という立場をとる。これにより、証明における制御演算子の使用が制限され、「shift/reset をもつ体系は直観主義論理である」という予想が導かれる。