

卒業研究を通して形式手法を学んで

比嘉健太^{†1} 河野真治^{†1}

卒業研究において形式手法を用いて問題を形式化した。この一年間でどのようなことを行なったか、どう感じたか述べる。

YASUTAKA HIGA^{†1} and SHINJI KONO^{†1}

1. 自己紹介

私は琉球大学工学部情報工学科に所属する四年次の学生だ。自動でプログラムのバグを見付ける機構が欲しいと思い、卒業研究はそのようなテーマに取り組んでいる。卒業研究において形式手法を用いて形式化することとなったのでその経緯と感想を述べたい。

2. 卒業研究と型と Agda

私の初期の研究テーマはプログラムから自動でバグを指摘する機構を作ることであった。

研究を進めるにつれ、プログラムのバグを見付けるには仕様やテストを記述する必要があることを知った。テストを記述するコストはおそらくプログラムを書くコストと同じかそれ以上かかる。あくまで私はプログラムのみでバグを見つけたかった。そのために、プログラムの変更に着目することによりバグを見付ける手法を提案することとなった。

そこで問題となったのがプログラムの変更の表現だった。この問題に対し、指導教員からプログラムに対する変更を Monad によって形式化するアイデアを頂いた。

Monad はプログラミング言語 Haskell では型クラスとして用意されており、型とは何かを知る必要があった。そこで、Proofs and Types を指導教員と読むこととなった。Proofs and Types では型と論理の対応について知ることができた。特に自然演繹は直感的に理解でき、対応する型は私の関心の対象となった。直

感的に理解できたのは型を持つプログラミング言語を記述したことがあるのも大きい。

Proofs and Types の例題は証明支援系言語 Agda によって記述した。Agda では項の定義とその書き換えによる等式変形を意識することとなった。Agda における証明は依存型で記述される。依存型を記述する言語の経験は無く、慣れるまでに時間が必要だった。あくまで等式の変形であることに気付くまでは依存型の趣旨が全く理解できなかった。

型の理論背景を知った後、次に Monad とは一体何か知る必要があった。Monad は圏論を元に計算機科学へ提案された表現であり、圏論を学ぶこととなった。Monad の理解に必要な圏や関手、自然変換などを学んだが、その解釈にも私は型を用いた。特に、Monad には満たすべき Monad 則が存在するが、依存型で表現してしまえば私にとって型に見えた。

型の理論背景と Monad を学んだ私は当初のアイデアであるプログラムに対する変更を Monad で表現することにした。Monad という指針が存在したため、プログラムの変更を Monad として定義することができた。特に、具体的にどのような則を満たすような型を定義すると良いか、という指針を得られたのは大きい。

3. 卒業研究を通して形式手法に思うこと

形式手法に関して、私の卒業研究を通して得た知見は2つある。

まず1つめは形式手法を直接用いるためには専門家が必須なことである。形式手法の提案とどのように形式化するかを指摘するには必須な知識が多く存在する。例えば、今回の指導教員が行なった指摘には論理や圏論やプログラミングなどに造形が深い必要がある。正直

^{†1} 琉球大学工学部情報工学科

Department of Information Engineering, University of the Ryukyus.

に言ってしまうと、指導教員の指摘が無ければ形式手法を用いる発想は出なかったであろう。

2 つめは、システムとして確立されているのなら利用しやすかったことである。型の理論的背景を理解せずとも、これまでにプログラムを書いた経験から型システムは問題無く利用できた。これはプログラムを書いた経験に基づくように思われる。

4. 形式手法を広めるには

形式手法を用いるには多くの知識が必須である。よって、前提知識を減らしたシステムとしてより手軽に利用する方が良いと考える。

例えば、Agda では証明はほぼ自ら記述するが、プログラムから証明が自動で導出できれば手軽に利用できる。型のように推論が可能であれば既存のプログラムにも適用しやすい。まずは制約が弱くとも導入しやすいシステムとして提供し、必要に応じて制約を強くするような手法はどうだろうか。

5. ま と め

私は卒業研究において形式手法を用いて問題を形式化した。その際に必要な知識は多く、形式化よりもその知識の修得に時間がかかってしまった。しかし、形式化する中で満たすべき条件が明確化されているのは処理を考える指針となった。

また、型に対する理論的背景が無くとも型システムは直感的に理解できた。導入が手軽なシステムとして普及させ、その後から強力な手法を適用するために制約を加えていくことで普及拡大を目指すのはどうだろうか。