

論 文 要 旨

Practicable type debugging for functional languages
(関数型言語のための実用的な型デバッグ作成手法の提案)
人間文化創成科学研究科 情報科学コース 対馬かなえ

本論文では、関数型言語のための実用可能な型デバッグ作成手法について述べる。Hindley-Milnerの型推論が導入されて以来、プログラマは型による恩恵を受けて来た。それは同時にプログラマと型エラーとの戦いの幕開けでもあった。プログラマによる型エラーの原因特定を手助けするために多くの手法が提案されており、それらの実装は実際にプログラマを助けてきたが、未だ多くの言語が型デバッグに関する実装を欠いている。実際、型デバッグのサポートがない言語では、型エラーの原因を探る際、プログラマはコンパイラのエラーメッセージを元に、自身で型を推論するなど時間を割いているという現状がある。このような状況は初級者にとって学習の障害になる上に、上級者にとってもデバッグを行う時間を必要とし、プログラマに負担を強いる。

この状況を打開するためには、実用的な型デバッグ手法が必要となる。しかし、「実用的な型デバッグ手法」とはかなり曖昧な表現であるため、まず我々はそれが満たすべき性質を提案した。この性質は大きく二つの種類に分けられる。ひとつ目は、型デバッグを実装する側に関する性質である。その性質は、型デバッグを多くの言語に適用することを目的に定められている。ふたつ目は、型デバッグを使用するユーザ側に関する性質である。その性質は、実際にユーザにとって使いやすい型デバッグになることを目的に定められている。これらの性質が合わさることで、実装面・利用面の両方から実用的な型デバッグが実現される。

本論文は大きく分けて二つの内容を扱う。

まずひとつ目の内容では、型デバッグを実装する側に関する性質に注目し、それらの性質を満たす型デバッグを作成する手法を提案する。これまでの型デバッグは、それぞれ特殊な型推論器を必要としており、それが障害となって多くの言語への適用が困難であった。我々はコンパイラの型推論器を使用することによって、型推論器の実装を必要としない型デバッグを提案した。我々の手法の利点は三点存在する。型推論器を実装せずコンパイラの型推論器を使用することによって、コンパイラの型推論器との齟齬が発生し得ない点、型推論器を再び作成する手間がかからない点、コンパイラの変更に強い点である。提案する型デバッグの主軸となる方法は、Chitilの対話的な型デバッグに基づいている。説明の流れとしては、まず提案の概要を説明し、その提案を単純型付きラムダ計算に適用する。その後、それをlet多相やオブジェクトに拡張し、この提案がどこまで拡張可能かをみる。

ふたつ目の内容では、型デバッグを使用するユーザ側に関する性質に注目し、それらの性質のうちひとつを満たす、重み付き型エラースライスという手法を提案する。これまでの型エラースライス

作成手法では、型デバッグの際に見る範囲を小さくすることが可能であった。これは実際にデバッグを行う際、ユーザの負担を軽減するために有効である。我々はこの方法を拡張し、スライスの各所に重みを付けた。ここでの重みとは、それぞれの箇所がどれくらいエラーの原因である可能性が高いかということを表す。よって、重み付き型エラースライスを使うことで、デバッグの際にエラーの可能性が高い箇所から質問をすることが可能となり、ユーザにとって使いやすい型デバッグを行うことが出来る。

このように実装面・利用面からそれぞれ型デバッグ作成手法を改良することによって、関数型言語のための実用的な型デバッグが実現された。これによって、型デバッグ機能を欠いた多くのプログラミング言語を対象とした型デバッグの実装が可能になる。型エラーの修正が容易になることは、型付きプログラミングの学習への貢献や、プログラム作成時の障害が減ることによる信頼性の高いソフトウェア開発などに繋がっており、そのための一歩が本研究によって為されたと考えている。