

# 初学者向けコンパイラ Helium のエラーメッセージの調査と改良

情報科学科 岩下 華穂

指導教員：大久保 弘崇

## 1 はじめに

プログラミング言語の処理系は、誤りのあるソースプログラムに対して、その誤りを修正するプログラマにとって有用なエラーメッセージを示すことができれば有用である。しかし、プログラマの技量によって適切なヒントは異なると考えられる。特にプログラミング初学者は知識の不足から情報の取捨選択ができないため、過剰な情報を提示したり、未学習の言語機能を提案するエラーメッセージは学習の障壁となる。

プログラミング初学者に向けてより適切なヒントを提示することを目標として、本研究の目的は、同様の方針で提案されている既存の処理系を対象として、それが誤りのあるプログラムに対して提示する情報の内容を調査することである。

## 2 Helium

Helium は、プログラミング言語 Haskell の初学者向けコンパイラである。初学者でも分かりやすいエラーメッセージが出力されるよう設計されている [1]。

例として、`sin 0.2`という式を誤って `sin .2`と入力した場合を考える。プログラミング言語 Haskell の代表的な処理系である GHC は、次のエラーメッセージを出力する。

```
<interactive>:1:1: error:
  * Non type-variable argument in the constraint: Num (a -> c)
    (Use FlexibleContexts to permit this)
  * When checking the inferred type
     it :: forall c a. (Floating c, Num (a -> c)) => a -> c
```

これは、Haskell の構文でこの式が `sin` と `2` を関数合成の中置演算子 `.` で結合していると解釈され、その解釈に基づいた上で、関数合成の左辺が浮動小数点数をとる関数 `sin` であるため、右辺は浮動小数点数を返す関数である必要があるがそうではないという型エラーを報告している。さらに Haskell 標準でない処理系の独自拡張を用いてこの問題を解決できる可能性があるというアドバイスを提示している。これらは Haskell や GHC の機能に十分習熟したプログラマにとっては有用であるが、初学者にはほぼ意味をなさない。

一方、Helium は次のエラーメッセージを出力する。

```
Warning: Function composition (.) immediately followed by number
Hint: If a Float was meant, write "0.2"
      Otherwise, insert a space for readability
Type error in infix application
expression      : sin . 2
operator        : .
type            : (a -> b) -> (c -> a) => -> c -> b
right operand   : 2
type            : Int
does not match : c -> a
```

後半は GHC と同様の型エラーを報告しているが、前半で Hint として、「浮動小数点数なら `0.2` と書く必要がある、関数合成な

らば可読性のためにピリオドの後に空白を入れることを推奨する」という、プログラマの意図に沿ったヒントを提示している。

このヒントを提示するために、構文解析の後では本来不要な字句レベルの情報を利用していることは注目に値する。

## 3 調査

Helium のエラーメッセージの調査と、その各々のエラーメッセージを出力するプログラムの作成を可能な限り行った。まず Helium のソースファイルから、エラーメッセージに関わる部分を抽出した。その内容からどのようなプログラムのミスであるかを推測し、そのミスに対応するプログラムを考え作成した。

例えば、`Helium4Haskell/helium/src/Helium/Utils/Texts.hs` には以下のようなメッセージが含まれている。

```
"Tab character encountered; may cause problems with the layout rule"
, "Configure your editor to replace tabs by spaces"
```

タブ文字の使用は問題の原因となるので推奨されないと警告している。これに対して次のプログラムを作成した。2 行目の `a` と `b` の間にタブを挿入している。

```
add :: Int -> Int -> Int
add a_____b = a + b
```

このプログラムを Helium でコンパイルすると、以下のようなメッセージが出力される。

```
(2,6): Warning: Tab character encountered; may cause problems with the layout rule
Hint: Configure your editor to replace tabs by spaces
Compilation successful with 1 warning
```

着目したメッセージが出力されていることが確認できる。

## 4 改良手法

調査から Helium の改良手法を提案する。調査により得られた Helium が想定するエラーパターンのプログラムを GHC でコンパイルし、そのエラーメッセージを比較する。比較により、GHCの方がわかりやすいと判断したエラーメッセージを「改良の余地がある Helium のエラーメッセージ」とし、その改良案を述べる。また調査中に得られた Helium の不具合を改良する案を述べる。

## 5 おわりに

本研究では、初学者向けの Haskell コンパイラである Helium のエラーメッセージの調査を行い、その改良案を提示した。今後の課題としては、改良案の実装と、改良後の Helium の評価が挙げられる。

## 参考文献

- [1] Bastiaan Heeren, Daan Leijen, Arjan van IJzendoorn, "Helium, for Learning Haskell", in Proc. of the 2003 ACM SIGPLAN Workshop on Haskell, pp. 62–71, 2003.