

Haskell を対象とした先行評価を模したトレースの実現手法

情報科学科 加藤知樹

指導教員：大久保弘崇

1 はじめに

C 言語などのメジャーなプログラミング言語では式の評価は先行評価で行われる。先行評価は関数呼び出しの際、引数の評価を行った後、関数本体の評価を行っていく。一方 Haskell は遅延評価である。遅延評価は関数呼び出しの際、関数本体の評価を行っていき、実際に引数が使われるまで引数の評価を遅らせる。そのため式がいつ評価されるか実行時までわかりづらい場合がある。つまりプログラムの意図しない動作の原因を特定するのが、ソースコードを見ただけでは難しくなる。そこでプログラムの実行順序を明らかにするトレースデバッグは有用な手法であると考えられる。しかし遅延評価に即したトレースを表示する場合、ある関数本体のトレースの中に引数のトレースが入り組み表示が複雑になるといった問題が考えられる。

そこで本研究では、Haskell の評価に対する理解を促すことを目的とし、遅延評価に即したトレースを先行評価でのトレースに疑似的に表示する手法を提案する。また提案手法を既存の Haskell トレースツールをもとに実装する。

2 先行評価を模したトレース

遅延評価のもとでは、「値が必要になるまで式を評価しない」という性質を利用した式や構造を作る。

例えばソースコード 1 のように if 式に相当する関数が作れる。

ソースコード 1 myIf.hs

```
myIf :: Bool -> a -> a -> a
myIf True  x _ = x
myIf False _ y = y

main = print (myIf True (1 + 2) (3 + 4))
```

一方、先行評価では評価される式の部分式はすべて評価される。そのため遅延評価のトレースを先行評価に即した表示にすると、本来評価されない式まで評価するように表示されてしまう。例えばソースコード 1 の場合、 $3 + 4$ を評価したように表示される。よって一般的に Haskell のトレース結果を先行評価に即した表現にはできない。そこで先行評価を模した表現では、評価されない式の疑似的な評価結果を特別な記号「□」で表す。

例えばソースコード 1 のプログラムの先行評価を模したトレースは図 1 のようになる。

```
> main
| > print (myIf True (1 + 2) (3 + 4))
| | > myIf True (1 + 2) (3 + 4)
| | | > 1 + 2
| | | < 3
| | | > 3 + 4
| | | < □
| | < 3
| < 10
< 10
```

図 1 先行評価を模したトレース

トレース中の記号について説明する。

- > は評価する式を表す。
- < は評価した結果を表す。
- | は関数呼び出しの深さを表す。

$3 + 4$ の評価結果として「□」と表示することで、この式が遅延評価のもとでは本来評価されない式であると示している。

3 Redex Trail

実装にあたって、Haskell の実行トレースを表示するツール Hat[1] を利用する。このツールは実行トレースの情報を記録するためのデータ構造として Redex Trail[2] を利用している。Redex Trail は、プログラム全体の評価を記録した有向グラフであり、Haskell の遅延評価での実行に関する詳細な情報を含んでいる。そこで提案手法に基づいて、先行評価を模したトレースを表示するツールを実現するにあたり、本研究では Hat が生成するこのファイルを利用することにした。

4 先行評価を再現する巡回

Redex Trail には評価の因果関係に関する情報が含まれているが、どのような順序で評価されたかという情報は含まれていない。したがって、ある評価戦略のもとでのトレースを表示するには、表示したい評価戦略を再現するように Redex Trail を巡回する必要がある。

「main を表すノードから始めて、注目しているノードが関数適用もしくは変数であれば、引数のノード、結果のノードの順に再帰的に Redex Trail を巡回し、未評価を表すノードもしくは正規形であれば、そのままノードを巡回の結果とする」とすれば先行評価を模した表現が得られる。

5 おわりに

本研究では、Haskell における遅延評価に即したトレース結果に比べ、先行評価を模したトレース結果のほうがわかりやすいという仮説のもとその手法を提案した。また Haskell における既存のトレースツールである Hat を使い、提案手法をツールとして実装した。

今後、提案手法が有用であるか評価していく。

参考文献

- [1] “The Haskell Tracer Hat.” <https://archives.haskell.org/projects.haskell.org/hat/>. (accessed:2018-12-12).
- [2] “The Definition of the Augmented Redex Trail.” <https://www.cs.york.ac.uk/fp/hat/Memos/augmentedRedexTrail.2.ps.gz>. (accessed:2018-12-12).