

コンパイラパーサによる構文解析器における言語指向のエラー報告

情報科学科 伊東 忠彦

指導教員：大久保 弘崇

1 はじめに

言語処理系の実装者はその構文を解析するパーサを作成する。このとき、Yacc[1] や Parsec[2] といったパーサ作成支援系を利用することが一般的である。パーサは、構文誤りのために受理できない入力を与えられたとき構文エラーを報告する。パーサ作成支援系が提供するエラー報告機能は、解析の継続が不可能になった位置、受理できなかったトークン、期待した入力などからなる。このエラー報告の内容そのものはパーサにバグがない限り正しいが、言語処理系の利用者に提示するものとしては十分でない場合がある。

パーサ作成支援系は与えられた構文規則の粒度で動作しエラー報告を行う。一方利用者は対象言語の構造の粒度で思考する。この粒度の違いが適切でないエラー報告につながる。利用者は構文誤りを修正する必要がある、エラー報告はその助けとなる情報を持つべきである。そのような視点でのエラー報告を本研究では「言語指向」と呼ぶ。

言語指向のエラー報告を行うパーサを作成するためには、パーサの作成者が適切な情報を構文に注釈付けしたり、構文エラー時の異常系処理を丁寧に実装する必要がある。また、パーサ作成支援系もその作業を支援するべきである。

本論文ではパーサコンパイラライブラリの Parsec に焦点をあて、ライブラリが提供するエラーコンパイラを利用した場合の、エラー報告の改善の方法を検証する。

2 Parsec

Parsec は、Haskell で記述されたパーサコンパイラライブラリである。小さなパーサを Haskell の関数として作成し、それらをコンパイラで組み合わせて複雑な構文を受理するパーサを構築する。

Parsec のマニュアルには、エラー報告をする際には `<?>`、`unexpected`、`fail` という 3 種類のコンパイラを用いるよう記述されている。その中の `unexpected` コンパイラは以下のような型であり、検出された時点で解析を失敗させ、エラーメッセージを引数にとった文字列に置き換えることができる。

```
String -> GenParser tok st a
```

3 実験

Parsec を用いて、エラー処理を行わず純粋に構文を受理するだけのパーサ (素のパーサと呼ぶ) を作成し、このパーサに `unexpected` コンパイラを用いて言語指向のエラー報告をさせる動作を追加し、結果を分析する。対象言語には picoC[3] を選択した。これは、本学の情報科学応用実験におけるコンパイラ作成実験で対象としている C 言語のサブセットである。

3.1 言語指向でないエラー報告

素の picoC パーサは、予期しないトークンを検出するとその時点で構文解析を中止し、中止の原因となったトークンと予期していたトークンを報告する。これは Parsec が提供するパーサ視点のエラー報告機能である。

リスト 1 式文を解析するパーサ部分

```
1 do { e <- exprparser
2   ; semi
3   ; n <- stmtparser
4   ; return (StmtExpr e n)
5 }
```

リスト 2 セミコロンの付け忘れがある picoC プログラム

```
1 int one() {
2   int a;
3   a = 1
4   return a;
5 }
```

例えば、picoC における式文はリスト 1 のような Haskell プログラムで解析される。リスト 2 のような、代入式においてセミコロンを付け忘れたソースを与えると、以下のようなエラー報告が得られる。

```
unexpected "r"
expecting operator or ";"
```

このメッセージは付け忘れたセミコロンではなく、それより後のエラーとは無関係のトークンに注目しており、構文誤りを修正する助けとして不十分である。

3.2 言語指向のエラー報告

こうした利用者の誤り方が予測できる場合に、言語指向のエラー報告を実現する方法を考える。3.1 節の例においては、リスト 1 の `semi` に

```
semi <|> unexpected "letter other than ; ;. 前の式に;
は付いていますか?"
```

と注釈付けすることで、中止の原因であるトークンを報告するだけでなく、足りないトークンの確認を含めたエラーメッセージに置き換える。これにより、利用者への助けとなる情報を持たせたエラー報告に近づける。

4 おわりに

本研究では `unexpected` コンパイラを利用したエラー報告の改善方法の検証をし、コンパイラを加える前と加えた後のメッセージを比較して考察を行った。今後の課題として、このエラー処理の差し込み方の妥当性を計測して比較する方法を確立することなどが挙げられる。

参考文献

- [1] Johnson, Stephen C. Yacc: Yet another compiler-compiler. Vol. 32. Murray Hill, NJ: Bell Laboratories, 1975.
- [2] Leijen, Daan. "Parsec, a fast combinator parser." (2001).
- [3] 情報科学応用実験指導書 情報システムコース, 愛知県立大学, 2016.