

COINS と LLVM の連携による例外処理対応とその評価

産一 傑^{1,a)} 川端 英之^{1,b)} 北村 俊明^{1,c)}

概要：コンパイラ基盤の一つである COINS[1] はレベルの異なる 2 種類の間中表現を持っており、各中間表現に対する最適化、変換モジュールを導入するだけで新たなコンパイラを構築することができる。我々は、COINS に新たな中間表現 EHIR を設計し、多様な言語のフロントエンドの構築を容易にした。本システムは、EHIR から HIR、あるいは LLVM IR に変換する時点で例外処理を埋め込むことにより、既存のバックエンドを活用している。本稿では、COINS に拡張を施した EHIR 及びバックエンドとしての LLVM の利用について述べる。

キーワード：言語処理系, COINS, LLVM, 例外処理

1. はじめに

コンパイラ基盤の一つである COINS[1] はレベルの異なる 2 種類の間中表現を持っており、各中間表現に対する最適化、変換モジュールを導入するだけで新たなコンパイラを構築することができる。我々は、COINS へ 2 返戻値法や setjmp 法といった複数の例外処理方式を用いた例外処理対応機能の拡張を行った [3]。現状では、LLVM[2] が提供する表引き法にも対応するために、LLVM との連携を試みている。本稿では、開発中のコンパイラ基盤の全体像について述べる。

2. 本システムの構成方式

本システムの構成を図 1 に示す。EHIR は、例外処理構文を表現するために新たに設計した高水

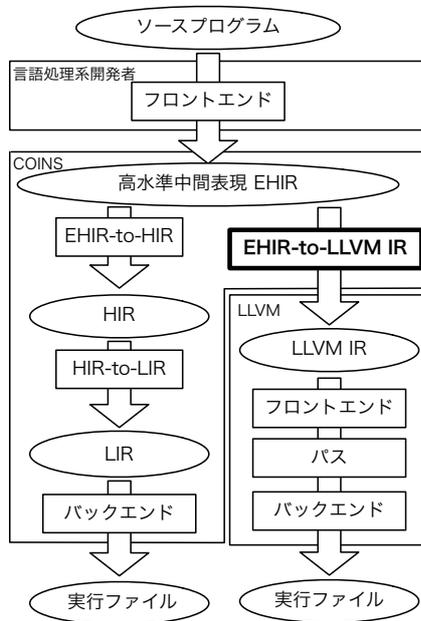


図 1 拡張コンパイラ基盤の全体像

¹ 広島市立大学
^{a)} mx66007@e.hiroshima-cu.ac.jp
^{b)} kawabata@hiroshima-cu.ac.jp
^{c)} kitamura@hiroshima-cu.ac.jp

準中間表現である。言語処理開発者は、EHIR を生成するフロントエンドを開発するだけで例外処理構文を持った新たな言語及び、言語処理系を開発することができる。また我々は、COINS へ 2 戻戻値法や setjmp 法といった複数の例外処理方式を用いた例外処理対応機能の拡張を行った。さらに COINS と LLVM を連携することによって、新たに表引き法による例外処理の実現を可能することができた。これによって、コンパイラとして本システムを利用するユーザに対して、利用するアプリケーションに適した例外処理方式の選択肢を提供することができると考えられる。

3. フロントエンドと EHIR

C++, Java, Python, Ruby などの言語に備えられている例外処理構文には以下のような機能がある。

- Java や C++ では、継承関係を持ったクラス型や基本型の例外のスロー又はキャッチができる。
- try-catch 文のような例外の検知、例外ハンドラ、最終処理の機能を持つ構文が存在する。

EHIR は、以上の機能を持つ例外処理構文を表現することができる高水準中間表現である。例えば図 2 は、EHIR で try-catch 文を表現したものである。Java や C++ に備えられている try-catch 文や Python の try 節、except 節の制御構造と同等の表現となっている。また、Java や C++ におけるクラス型のための表現を EHIR へ取り入れることによって、継承関係を持つ型による例外処理の表現も可能としている。

```
{ try void
  (... <labelDef _lab1 >
  (... ( call ...< subp ... MayThrow >)... )
  ( list ...
  ( catch void
  < var int i >
  (... <labelDef _lab2 >
  ( catch body ...))) ... )
```

図 2 EHIR の表現例

4. EHIR とバックエンド

EHIR で表現された例外処理構文による制御構造を実現する例外処理方式のうち、2 戻戻値法、setjmp 法については、EHIR を HIR に変換する

時点で取り込むことができる。2 戻戻値法では戻り値を、setjmp 法では setjmp()/longjmp() の呼び出しにより、例外処理機能を埋め込むことができる。一方で表引き法は、LLVM IR に変換することにより、図 3 のようにして例外処理機能を埋め込んでいる。図 3 に示されるように LLVM では、中間言語及び組み込み関数と Itanium C++ ABI によって表引き法による例外処理を実現している。本システムでは、EHIR で表現された例外処理構文を図 3 の様な LLVM IR へ変換することによって、表引き法による例外処理の実現を試みている。

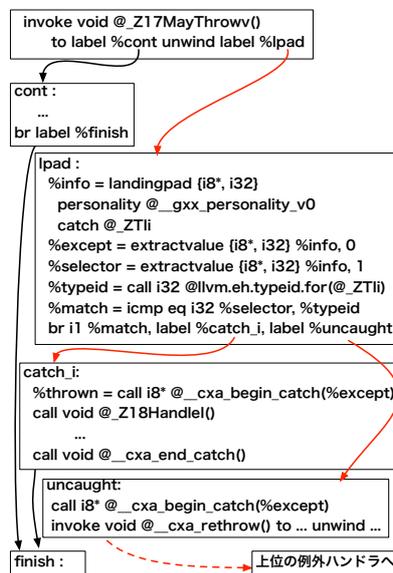


図 3 LLVM IR における例外処理

5. まとめ

本稿では、多様な例外処理方式に対応させたコンパイラ基盤の構成について述べた。COINS に新たに導入した中間表現 EHIR により、多様な言語のフロントエンドが容易に構築可能である。また、EHIR から HIR、あるいは LLVM IR に変換する時点で例外処理を埋め込むことにより、既存のバックエンドが活用可能である。

参考文献

- [1] COINS コンパイラ・インフラストラクチャ: <http://coins-compiler.sourceforge.jp/>
- [2] The LLVM Compiler Infrastructure: <http://llvm.org/>
- [3] 産一 傑, 川端 英之, 北村 俊明: COINS による例外処理機構を持つ言語処理系開発支援, 日本ソフトウェア学会第 30 回大会 講演論文集: ソフトウェア論文, Sep. 2013.