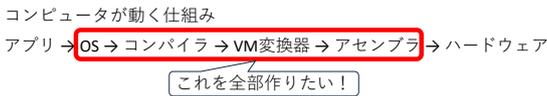


0から作るコンパイラ ～自作コンピュータ(ホンモノ)を作る～

背景

コンピュータが動く仕組みを知りたい
プログラミング言語を動かすためのコンパイラを書きたい



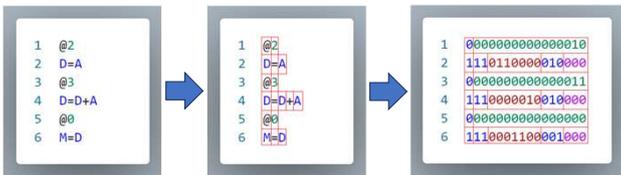
Nand2Tetris

書籍「コンピュータシステムの理論と実装^{[1][2]}」の通称。
NANDゲートからCPUを作り、それを動かすアセンブリや高級言語のコンパイラを書いていくという内容で構成されている。



アセンブラ

コンピュータの頭脳・CPUを動かすための言語・アセンブリを機械語に変換するプログラム。
アセンブリは、機械語と1対1で対応する単語列から成り、アセンブラはこれをトークン列に分割し、機械語に変換する。



VM (仮想マシン)

物理CPU上に仮想的に実装するコンピュータのことであり、ここではスタックを用いたものを実装する。

コンパイラ

高級言語をVMの言語に変換するプログラムである。ここでは、独自の高級言語Jackのコンパイラを作成した。
アセンブラやVM変換器などと同様に、構文解析やコード生成を行う必要はあるが、その中でも特に難しかった。



```

void CompilationEngine::compileWhile(){
  ofs << "whileStatement>";
  std::string label_str = newWhileLabel();
  std::string while_exp = "while_exp" + label_str;
  std::string while_end = "while_end" + label_str;

  // write VM
  vmWriter.writeLabel(while_exp);

  // "while"
  compileKeyword();
  // '{'
  compileSymbol();
  // expression
  compileExpression();
  // '}'

  // write VM
  vmWriter.writeArithmetic(Command::NOT);
  vmWriter.writeLabel(while_end);

  compileSymbol();
  // ';'
  compileSymbol();
  // statements
  compileStatements();
  compileSymbol();

  // write VM
  vmWriter.writeGoto(while_exp);
  vmWriter.writeLabel(while_end);
}
  
```

図1.設計したコンパイラのクラス図 図2.実装したコンパイラのソースコード (抜粋)

文脈自由文法

生成文法において、生成規則が全て $V \rightarrow \omega$ であるような形式文法のこと^[3]。非終端記号 V が 終端記号と非終端記号の並び ω により表される。

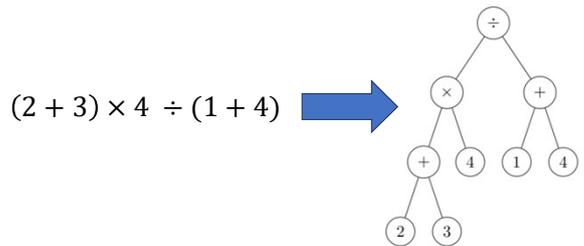
具体例: Jack の whileStatement

```

statements → statement*
statement → letStatement | whileStatement | ifStatement | returnStatement | doStatement
whileStatement → 'while' '(' expression ')' '{' statements '}'
expression → term (op term)*
  
```

構文木

構文解析の経過や結果を木構造で表したものの^[3]。構文解析とは、ある言語において、その形式文法に従って記号の文字列を分析する手続きのことである。



成果

本企画で制作したC++のソースコードやクラス図などは、GitHub(<https://github.com/nao109/nand2tetris>)で公開している。C++のコンパイラにはg++を用いた。

まとめ

アセンブラやVM変換器、コンパイラをC++で制作した。クラス図や構文木を書き、実際にプログラムを実装したことで、コンピュータの動作原理を深く理解できた。

最後に本企画のアドバイザーを務めていただいている、知能情報メディア課程の芝公仁助教には、プログラム設計や、このポスター制作などで様々なアドバイスをいただいた。ここに深謝の意を表する。

今後の展望

JackをARMアーキテクチャの命令に変換するプログラムをC++で書き、Raspberry Pi上にJackを実装したいと考えている。

参考にさせていただいたもの

- [1] Noam Nisan, Shimon Schocken, 齊藤康毅 . コンピュータシステムの理論と実装
- [2] From Nand to Tetris . <https://www.nand2tetris.org/>
- [3] Wikipedia . <https://www.wikipedia.org/>
- [4] havivha . <https://github.com/havivha/Nand2Tetris/>
- [5] Rui Ueyama .低レイヤを知りたい人のためのCコンパイラ作成入門