

主専攻実験[S-8] 関数プログラミング

課題 8-3

情報科学類 202113564 三村潤之介

上記の説明を参考にして、ミニ OCaml 言語から CAM の機械語へのコンパイラを実装しなさい。

github リポジトリ上の 11 8-1/8-3.ml にて compile 式,completop 式(引数を exp 式のみもち、空環境を入れて compile 式を実行する式)として実装した。

https://github.com/mimunojun/functional_prog/tree/master/11%208-1

コンパイラが正しく変換できていることを示す例を与えて実行しなさい。

資料にある簡単な和の例を実行した。exp 式は以下ようになる。

```
let exp1 = Let("x",IntLit(1),Let("y",IntLit(2),Plus(Var("x"),Var("y"))))
```

completop を用いて上記の exp1 をコンパイルした。

```
# completop exp1;;
```

```
- : cam_code =
```

```
[CAM_Ldi 1; CAM_Let; CAM_Ldi 2; CAM_Let; CAM_Access 0; CAM_Access  
1; CAM_Add;
```

```
  CAM_EndLet; CAM_EndLet]
```

資料と同様に CAM 命令列へ変換できた。

変換されたコードを、CAM の実装にかけて実行し、正しい答えが返ってくるかを試しなさい。

以下のような再帰関数を含むコードを与えた。

```
let rec f x =
```

```
  if x=0 then 1
```

```
  else 2 + rec (x-1)
```

```
in (f 3)
```

exp 式にすると以下ようになる。

```
let exp2 = LetRec("f", "x",  
                  If(Eq(Var("x"),IntLit(0)),  
                     IntLit(1),  
                     Plus(IntLit(2),App(Var("f"),Plus(Var("x"),IntLit(-1))))  
                  ),  
              App(Var("f"), IntLit(3)))
```

課題 8-2 で実装した transtop(CAM の遷移を行う実装)及び completop を用いて上記の exp1 をコンパイルし、CAM で実行した。

```
# transtop (completetop exp2);;  
- : cam_value = CAM_IntVal 7
```

以上から、正しく実行できていることがわかる。