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,compiletop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式は以下のようになる。

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

compiletopを用いて上記のexp1をコンパイルした。

# compiletop 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式にすると以下のようになる。

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compiletopを用いて上記のexp1をコンパイルし、CAMで実行した。

# transtop (compiletop 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。