

## プログラム言語論 課題 2

氏名 (学籍番号) : 春名航亨

(201811528 )

課題 2-1 (a) 右結合

(b) 優先度は  $! > \&\& > ||$  の関係になっている

課題 2-2 (a) `for (i = 0; i < 10; i++)`

`for (j = 0; j < 10; j++)`

`m = i+j;`

`r = sqrt(i*i+j*j);`

`if (m < 3 || 10 < m) {`

`printf("r=%f\n", r);`

`}`

(b) 文法の他に意味を指定することが必要である。

例えば、 $[0-9]^+$ の文字をプログラムに用いるという文法を決めても、それが「数値」を示すものだという意味も併せて決める必要がある。

(c) 与えた BNF などの記述を解釈して、構文解析を行うパーサやインタプリタ、コンパイラを出力する処理系のこと。実装例: Flex, lex, re2c など

(d) ユーザーがプログラムをテキストではなく、関係を表す矢印や線、円弧などをグラフィカルに操作して作成するプログラミング言語のこと。

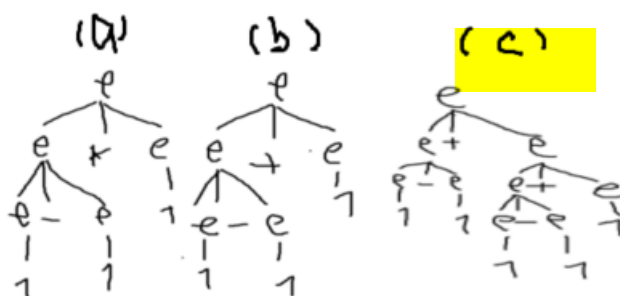
実装例: Blender, Scratch, Pure Data など

課題 2-3 (a) コード生成器を変えればよい。

C 言語コンパイラはコード解析器を用いて中間言語であるアセンブリを機械語に変換しているが、今回の目的のために、変換先を任意の CPU の命令セットを用いた機械語に変更すればよい。

(b) 字句解析器と構文解析器…yacc や lex といった構文解析器生成系を用いれば、新言語の BNF のような字句解析器と構文解析器の要素を持ったコードを与えて実行するだけで新言語を C 言語に変換できるため。

課題 2-4



## プログラム言語論 課題 2

氏名 (学籍番号) : 春名航亨

(201811528 )

- 課題 2-5 (a)  $\langle \text{seq} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{alpha} \rangle \mid \langle \text{digit} \rangle \langle \text{seq} \rangle \mid \langle \text{alpha} \rangle \langle \text{seq} \rangle$   
 $\langle \text{var} \rangle ::= \langle \text{alpha} \rangle \mid \langle \text{alpha} \rangle \langle \text{seq} \rangle$   
(b)  $\langle \text{dseq} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{digit} \rangle \langle \text{dseq} \rangle$   
 $\langle \text{real} \rangle ::= \langle \text{dseq} \rangle "." \langle \text{dseq} \rangle \mid \langle \text{dseq} \rangle "." \mid "." \langle \text{dseq} \rangle$

- 課題 2-6 (a) `$ for f in calc1.y y.tab.c;do eval 'wc -' {c,l}" $f;"done|tr ¥¥n /`  
`902 calc1.y/57 calc1.y/47068 y.tab.c/1576 y.tab.c/`  
これより、  
- バイト数でみると 902B の calc1.y は y.tab.c の 47068B と比べると小さい  
- 行数で見ると 57 行の calc1.y は y.tab.c の 1576 行と比べると小さい  
(b) メリット: フルスクラッチの手書きで書くより生成系の方が BNF などを書いて比較的簡単に高速に構文解析器が作れるので、作業効率は良い。  
デメリット: 機械的に生成された構文解析器のコードは人の手でデバッグされたコードより冗長で、簡潔でなく、メモリ効率も余り良くない。

- 課題 2-7 (a)  $\{y\}$

(b)  $(\lambda x. \lambda y. xy)(\lambda x. xy)$   
 $\rightarrow \alpha (\lambda x. \lambda y. xy)(\lambda a. a y)$   
 $\rightarrow \alpha (\lambda x. \lambda y. xy)(\lambda a. a z)$   
 $\rightarrow \beta \lambda y. (\lambda a. a z)y$   
 $\rightarrow \beta \lambda y. y z$

- 課題 2-8 (a)  $(\lambda f. \lambda g. f(g\ 1))(\lambda x. x+4)(\lambda y. 3-y)$   
 $\rightarrow \beta (\lambda g. (\lambda x. x+4)(g\ 1))(\lambda y. 3-y)$   
 $\rightarrow \beta (\lambda x. x+4)((\lambda y. 3-y)\ 1)$   
 $\rightarrow \beta ((\lambda y. 3-y)\ 1)+4$   
 $\rightarrow \beta 3-1+4$   
(b)  $(\lambda f. \lambda g. f(g\ 1))(\lambda x. x+4)(\lambda y. 3-y)$   
 $\rightarrow \beta \lambda f. \lambda g. f(g\ 1)((\lambda y. 3-y)+4)$   
 $\rightarrow \beta \lambda f. f((\lambda y. 3-y)+4)\ 1$   
 $\rightarrow \beta (\lambda y. 3-1)+4$   
 $\rightarrow \beta 3-1+4$

- 課題 2-9 (a)  $(\lambda \text{compose}. (\lambda h. \text{compose}\ h\ h\ 3)\ \lambda x. x+x)\ \lambda f. \lambda g. \lambda x. f(g\ x)$   
 $\rightarrow \alpha (\lambda \text{compose}. (\lambda h. \text{compose}\ h\ h\ 3)\ \lambda x. x+x)(\lambda f. \lambda g. \lambda z. f(g\ z))$   
 $\rightarrow \alpha (\lambda c. (\lambda h. c\ h\ h\ 3)\ \lambda x. x+x)(\lambda f. \lambda g. \lambda z. f(g\ z))$

## プログラム言語論 課題 2

氏名 (学籍番号) : 春名航亨

(201811528 )

---

$\rightarrow \beta (\lambda h. (\lambda f. \lambda g. \lambda z. f(g\ z))\ h\ h\ 3)\ \lambda x. x+x$   
 $\rightarrow \beta \lambda f. \lambda g. \lambda z. f(g\ z)\ (\lambda x. x+x)\ (\lambda x. x+x)\ 3$   
 $\rightarrow \beta \lambda g. \lambda z. (\lambda x. x+x)\ (g\ z)\ (\lambda x. x+x)\ 3$   
 $\rightarrow \beta \lambda z. (\lambda x. x+x)\ ((\lambda x. x+x)\ z)\ 3$   
 $\rightarrow \beta (\lambda x. x+x)\ ((\lambda x. x+x)\ 3)$   
 $\rightarrow \beta (\lambda x. x+x)\ (3+3)$   
 $\rightarrow \beta 3+3+3+3$

課題 2-10 (a)  $f := (\lambda g. g\ g)$

(b)  $f(f) = (\lambda g. g\ g)\ (\lambda g. g\ g) \rightarrow \beta (\lambda g. g\ g)\ (\lambda g. g\ g) \rightarrow \beta \dots$

以上と同じことが 6 行目  $x = f(f);$  の箇所で行われ、 $f$  は終了することなく  $x$  にいつまでも返り値を渡すこともないので次行の処理が進まない。