

言語プロセッサ 定期試験練習問題

実施日：平成25年1月21日(月)

担当：亀田弘之(持ち込み不可)

問題 1 以下の文章中の(1)~(4)に入れるべき最も適切な用語をそれぞれ答えなさい。

[全部正解で5点]

『Compiler とは (1) で書かれたプログラムを 機械向き言語に(2)するためのプログラムであり、その処理は、読み込み、字句解析、(3)、意味解析、中間コード生成、(4)、コード生成等のフェーズからなる。』

(出題意図：コンパイラの目的、処理概要を理解している。)

問題 2 次の中置記法で書かれた数式を後置記法で表現せよ。[10点]

$$(x + y) \times z - c \div (a + b)$$

(出題意図：数式の記法の意味を理解し、相互変換できる。)

問題 3 次の3つの規則は EBNF で書かれている。以下の問(1)と(2)に答えよ。[15点]

(規則 a) Program ::= ProgramHeading ";" Block ".".

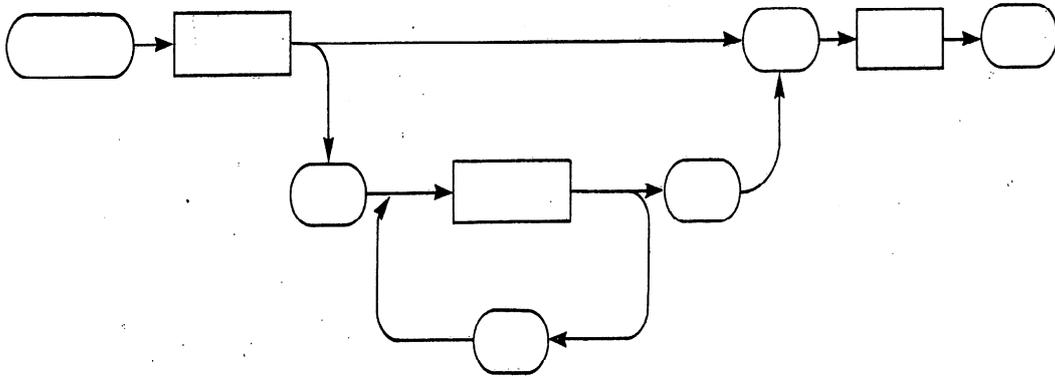
(規則 b) ProgramHeading ::= "program" Identifier [ProgramParameterList].

(規則 c) ProgramParameterList ::= "(" Identifier ")".

(1) EBNF を日本語では一般に何記法というか。カタカナで答えよ。[5点]

(2) 下図は上記3つの規則を1つの構文図で表現したものである。図を完成させよ。[10点]

Program



(出題意図：BNF と構文図の意味と相互関係を理解している。)

問題 4 次の正規表現 について問(1)~(3)に答えよ。[20点]

$$\alpha = (1|22)^* a(333|4444)^*$$

(1) 正規表現 が表す文字列の内、文字列長が1のものをすべて列挙せよ[5点].

(2) 正規表現 が表す文字列の内、文字列長2のものすべて列挙せよ[5点].

(3) 正規表現 が表す文字列の内、文字列長が4のものすべて列挙せよ[10点].

(出題意図：正規表現を解釈することができる。)

問題 5 右記の文法 G について問 (1) ~ (9) に答えよ。[30 点]

- (1) 開始記号はどれか。[1 点]
- (2) 終端記号をすべて列挙せよ。[3 点]
- (3) 非終端記号をすべて列挙せよ。[2 点]
- (4) $\text{First}(\quad)$ を求めよ。[2 点]
- (5) $\text{First}(i)$ を求めよ。[2 点]
- (6) $\text{First}(F)$ を求めよ。[5 点]
- (7) $\text{First}(E)$ を求めよ。[5 点]
- (8) $\text{Follow}(E)$ を求めよ。[5 点]
- (9) $\text{Follow}(F)$ を求めよ。[5 点]

文法 $G=(V, N, E, P)$

$$P = \{ \begin{array}{l} E \rightarrow TE', \\ E' \rightarrow +TE' \mid \varepsilon \\ T \rightarrow FT' \\ T' \rightarrow *FT' \mid \varepsilon \\ F \rightarrow (E) \mid i \end{array} \}$$

(出題意図: First、Follow を理解し、実際に求めることができる。)

問題 6 次の問 (1) ~ (4) の内、どれか 1 つを選び解答せよ。[20 点]

- (1) コンパイラの存在意義について 50 字程度で説明せよ。
(ヒント: コンパイラがないとどんな点が不便か考えよ)
- (2) コンパイラ型言語とインタプリタ型言語の得失に関する下記の表を完成させよ。

言語の種類	長所	短所	言語の例
コンパイラ型言語			Java, C, C++, C#, Fortran, Cobol, Ada, Pascal, Algol
インタプリタ型言語			BASIC, Lisp, Perl, PHP, Python, OCaml, Haskell,

- (3) コンパイラの処理の 1 つに最適化がある。例をあげて具体的にその処理内容を説明せよ。
- (4) 次の用語 (ア) ~ (カ) の中から 1 つ選び、50 字程度で簡単に知るところを述べよ。
可能ならば例をあげながら説明すること。

- (ア) LL(1)文法
- (イ) 括りだしと左再帰削除
- (ウ) トークン
- (エ) 四つ組
- (オ) 構文木
- (カ) Flex と Bison

(以上)