

# ソフトウェア演習 1

## 第 2 回レポート課題

J2200071 齊藤 隆斗

2024 年 5 月 21 日

### 1 課題 3.7&3.8+ 応用課題 3.9

#### 課題 3.7

算術式の構文解析ルーチンを参考にして正規表現の構文解析ルーチンを作成し、入力正規表現の構文木を表示するプログラムを作成せよ。正規表現の\*は、算術式の符号の場合と異なり、後置演算子であることに注意する必要がある。

#### 課題 3.8

引数 -d0, -d1, -d2 を指定することによって、入力正規表現、トークン列、構文木の 3 つをそれぞれ表示できるように main 関数を変更せよ。

#### 応用課題 3.9

通常の grep プログラムでは、連結の '.' は省略されるのが普通である。そのような場合、上の方法で正規表現の構文解析がうまく行く確かめよ。うまくいかないとすれば、どのように方針を修正すれば良いだろうか？

### 2 実行結果

#### 課題 3.7 が正常に動作することを確認

実行例 1: テキストの例について確認

```
$ ./kadai2 'a.\|.b'
```

```
LETTER(a)
```

```
CONC
```

```
LETTER(|)
```

```
CONC
```

```
LETTER(b)
```

実行例 2: テキストの例について確認

```
$ ./kadai2 '(a|b*)|c'
```

```
    LETTER(a)
```

```
    VERT
```

```
        LETTER(b)
```

```
    AST
```

```
VERT
```

```
    LETTER(c)
```

実行例 3: 追加の例についても確認

```
$ ./kadai2 '((a|b)*|\e).\0'
```

```
    LETTER(a)
```

```
    VERT
```

```
        LETTER(b)
```

```
    AST
```

```
    VERT
```

```
        EPSILON
```

```
CONC
```

```
    EMPTY
```

実行例 4: 右結合であることを確認

```
$ ./kadai2 'abcd'
```

```
    LETTER(a)
```

```
CONC
```

```
    LETTER(b)
```

```
    CONC
```

```
        LETTER(c)
```

```
    CONC
```

```
        LETTER(d)
```

### 課題 3.8 が正常に動作することを確認

実行例 5: d0 オプションの動作について確認

```
$ ./kadai2 -d0 '(a|b*)|c'
```

```
(a|b*)|c
```

実行例 6: d1 オプションの動作について確認

```
$ ./kadai2 -d1 '(a|b*)|c'
```

```
LPAR
```

```
LETTER(a)
```

```
VERT
LETTER(b)
AST
RPAR
VERT
LETTER(c)
EOREG
```

実行例 7: d2 オプションの動作について確認

```
$ ./kadai2 -d2 '(a|b*)|c'
    LETTER(a)
    VERT
        LETTER(b)
    AST
VERT
    LETTER(c)
```

実行例 8: 予期しないオプションに対するエラー処理について確認

```
$ ./kadai2 -d3 '(a|b*)|c'
invalid option
```

## 2.1 応用課題 3.9 が正常に動作することを確認

実行例 9: テキストの例について確認

```
./kadai2 -d2 'ab*c'
    LETTER(a)
CONC
```

```
    LETTER(b)
    AST
    CONC
    LETTER(c)
```

実行例 10: 追加の例について確認

```
$ ./kadai2 -d2 'a(\\0|\\e)*|b\\e'
    LETTER(a)
    CONC
        EMPTY
    VERT
        EPSILON
```

AST  
VERT  
LETTER(b)  
CONC  
EPSILON

### 3 プログラムの流れ

### 4 考察