問題記述と形式化 (中井担当分)

レポート課題

2017年度担当: 中井央

2017.12.18

課題内容

課題にあたっての注意事項

本レポート課題では、作成したプログラムとそれを 説明した文書の双方を提出する必要がある。プログラ ム作成についての注意事項を以下に述べる。

以下の項目を遵守すること。

- プログラムは Ruby で作成すること。
- プログラム作成にあたっては、この課題全体のディレクトリ (フォルダ) の名前を各自の学籍番号とすること。ディレクトリ名には半角数字を用いること。
- 全体ディレクトリの中に課題番号をディレクトリ 名とするサブディレクトリを設置すること。
- 各課題番号ディレクトリ内にはその課題のために必要なファイルのみ作成すること。また、Readme.txtというファイルを作成し、このディレクトリ内のファイルの一覧を含め、各ファイルの簡単な説明 (例:構文解析器クラスが定義されたファイル) を含めておくこと。Readme.txtの中には、それ以外に必要なことがらがあれば、それらを含めてよい。
- 各ファイル名は、指定があるものはその指示に従 うこと。
- 課題を満たしていることを確認するためのデータ ファイルや出力結果が納められたファイルは適宜、 当該ディレクトリへ含めてよい。

課題の概要

JSON というデータ形式がある。以下のステップを踏んで、課題のプログラムを作成せよ。 作成は講義で述べた方法に基づいて行うこと。 各ステップの詳細は後述する。各ステップの配点は次の通りである。中井担当分を 100 点満点で採点し、最終評価は、鈴木先生担当分と中井担当分を同一比率にした後の合計点にて行う。

- 1. JSON を入力として受け付ける構文解析器の作成 (40 点)
- 2. 構文解析木を作らずに構文解析木を表示するため の dot 形式の出力をする構文解析器の作成 (20 点)
- 3. 構文解析木を作成し、各ノードクラスにてアクションを定義し、構文解析木を表現する dot 形式を出力するプログラムの作成 (20点)
- 4. 構文解析木を作成し、Visitor を用いることで、構 文解析木を表現する dot 形式を出力するプログラ ムの作成 (20 点)

いずれも動作しない中途半端なプログラムの場合は 採点の対象としない。

以下、それぞれの詳細を記す。

JSON の構文解析器の作成

この講義で述べた方法に基づいて JSON 形式の入力を受け付ける構文解析器を作成せよ。JSON の文法を図1に与える。字句解析クラスのプログラムを図3

与えておく。適宜変更して使用してよい。JSON のサンプルを図2に示す。JSON そのものについては各自で調べられたい。

```
json : object
    | array ;
object : LBRA members RBRA ;
members :
       | memberlist ;
memberlist : pair (COMMA pair)* ;
pair : STRING COLON value ;
array : LBRACKET elements RBRACKET ;
elements :
        | elementlist ;
elementlist : value (COMMA value)*;
value : STRING
     | INT
     | FLOAT
     | TRUE
     | FALSE
     I NULL.
     | object
     | array
```

図 1: JSON の文法

図 2: exampleO.json

アクションによる dot 形式の出力

ここではアクションによって、dot 形式の出力をして、構文解析木を得られるようにすること。dot 形

式の出力については、第2回講義メモ(2)で示した rdp2graph.rb を参考にすること。なお、文字列や値は、それがわかるようにノード内に表示できるようにすること。出力例を図4に示す。

ツリーの作成とその上での操作

課題3および4について述べる。

どちらの課題もまず、ツリーを作る必要がある。ノードの作成にあたっては、各文法記号がノード種別となるようにクラスを作成すること。

```
case @line
class JsonLexer
 def initialize(f)
                                                             when /\A\:/
                                                               yield $&
   @srcf=f
   @line=""
                                                               token = :colon
   @linenumber=0
                                                             when /\A\[/
                                                               yield $&
 end
                                                                token = :lbracket
                                                             when /\A\l/
 attr_reader :linenumber
                                                               yield $&
 def lex()
                                                               token = :rbracket
   if /^s+/=^c Oline
                                                             when /\A\{/}
                                                               yield $&
     @line = $'
                                                               token = :1bra
   while @line.empty? do
                                                             when /\A\}/
     @line = @srcf.gets
                                                               yield $&
     if @line == nil
                                                               token = :rbra
                                                             when /\A\,/
       return :eof
     end
                                                               yield $&
                                                             token = :comma
when /\"[^"]*\"/
     @linenumber += 1
     if /\A\s+/ = \cite{Max} @line
       @line = $'
                                                               yield $&
     end
                                                               token = :string
    end
                                                             when /A[+-]?\d+\.\d+([eE][-+]?[0-9]+)?/
                                                               yield $&
                                                               token = :float
                                                             when /\A[+-]?\d+/
                                                               yield $&
                                                               token = :int
                                                             when /\Afalse/
                                                               yield $&
                                                               token = :false
                                                             when /\Atrue/
                                                               yield $&
                                                               token = :true
                                                             when /\Anull/
                                                               yield $&
                                                               token = :null
                                                             when /\A\s/
                                                               # ignore
                                                               token = :whitespace
                                                             when /\A\S/
                                                               # ignore
                                                               token = :other
                                                             end
                                                             @line = $'
                                                             puts "matched is #{token}(#{$&})"
                                                             return token
                                                           end
                                                         end
```

図 3: json_lexer.rb

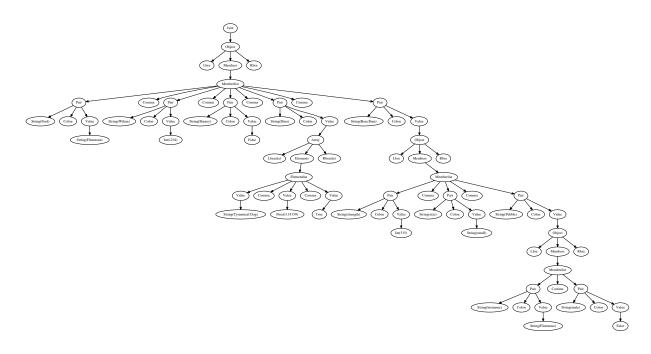


図 4: dot による構文解析木の例

レポートの提出方法等

レポートは次の要件に基づいて manaba にて提出すること。

- 次の2つのものを提出すること。
 - 作成したプログラム一式 ディレクトリ名等の詳細は、「課題にあたっ ての注意事項」で述べたことに従うこと。プログラム一式は、この課題のトップディレクトリ以下を zip もしくは tar(+gzip) で固めたものとする。この2つ以外の形式では受け付けない。

ファイル名 (拡張子の前) は学籍番号とする。ファイル名には半角数字を用いること。

以下に述べる様式に基づいた課題の説明等の 文書 文書の形式は PDF とする。ファイル名(拡

文書の形式は PDF とする。ファイル名 (拡張子の前) は学籍番号とする。ファイル名には半角数字を用いること。

文書の要件

文書は、次の要件を満たすように作成すること。

- 表紙だけで一ページとしないこと。
- この授業タイトル、担当者名 (中井央)、学籍番号、 氏名、提出日を記すこと。
- 各課題で作成したプログラムの説明 課題のそれぞれについて、各行単位などの細かな 説明は 不要 である。それぞれの課題について、 説明すべきポイントをしぼって、その題意を満 たすための設計、コーディング上の工夫、失敗し た考え方(あれば)等、どのように課題に取り組 み、解決したかがわかるように記述せよ。また、 どのように確認をしたかも記すこと。
- 感想

感謝の言葉は不要である。ここで記述して頂きたいのは、内容についての具体的な感想である。

〆切

2018年1月3日(水) 12:00 までに提出すること。

read.meファイルを作る ファイルとその説明 rde2graph