

Packrat parsing in Ruby

Shugo Maeda

2013-04-20

What's packrat parsing?

**"Top-down parsing with backtracking –
except:**

Uses memoization to achieve linear parse time"

by Bryan Ford

再帰下降構文解析

生成規則に対応した関数の相互再帰で実装

予測的パーサ (predictive parser)

- 入力から先読みしてどの候補を選択するか決定する
 - 例: 次のトークンが '(' だったら grouping-expression
- 速い
- 解析できる文法に制限がある

後戻りパーサ (backtracking parser)

- 複数の候補を順に試し、後戻りしながら構文解析を行う
- 遅い
- 解析できる文法に自由度がある

Packrat parser

後戻りパーサ

入力の大きさに対して線形時間で解析可能

- メモ化によって実現

Memoization

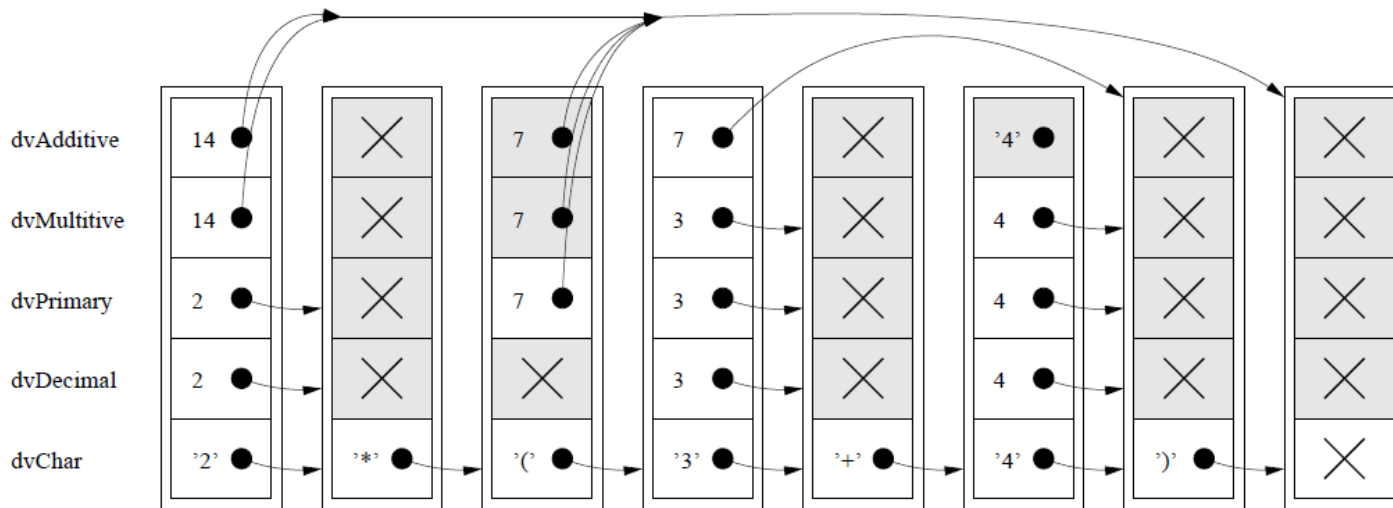


Figure 3. Illustration of Derivs data structure produced by parsing the string $'2 * (3 + 4)'$

From "Packrat Parsing: Simple, Powerful, Lazy, Linear Time" by Bryan Ford

How to write it in Ruby?

Rubyでやってみたのでコードで説明します

Haensel

<https://github.com/shugo/haensel>

- Packrat parser combinator library for Ruby
- ヘンゼルとグレーテルのパンくずの話に似てるかなと思ったけどよく考えたらそうでもないような

Example

```
@arith_grammar = Haensel::Grammar.new(:additive) {  
  define :additive,  
    multitive.bind { |x|  
      char(?+).bind {  
        additive.bind { |y|  
          ret x + y  
        }  
      }  
    } /  
  multitive
```


Conclusion

以下のような知見が得られた

- 確かに速くなった
- monadic parser combinatorがカッコいい
- Rubyでも末尾呼び出しの最適化ができる
- でもProcには効かない
- Immutable::Promise#lazyで回避できる
- でも結局スタックが足りない

Haskellでいいんじゃないね？