テキスト処理 第10回 (2008-06-24)
cap_exact で match_include

田中哲
産業技術総合研究所
情報技術研究部門
akr@isc.senshu-u.ac.jp
http://staff.aist.go.jp/tanaka-akira/textprocess-2008/

レポート

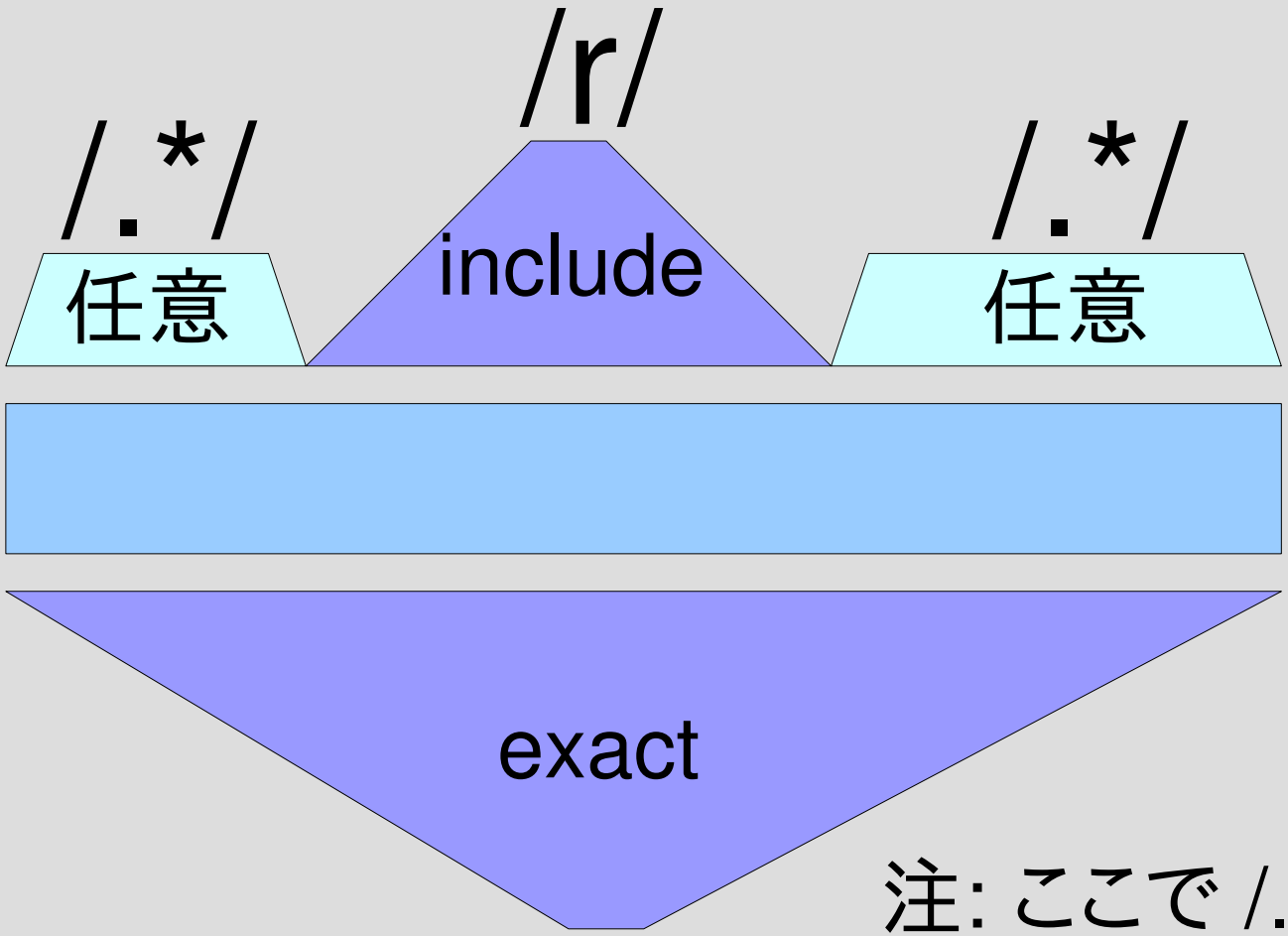
- cap_exact を用いて match_include を定義せよ
- そうやって実装した match_include と講義で説明したものに挙動の違いが有れば具体例をあげて違いを述べよ

- ユニットテストを提供するので、実装したらテストして確認すること
- 〆切 2008-06-24 12:00
- RENANDI
- 拡張子が txt なテキストファイルを望む

match_include の実行例

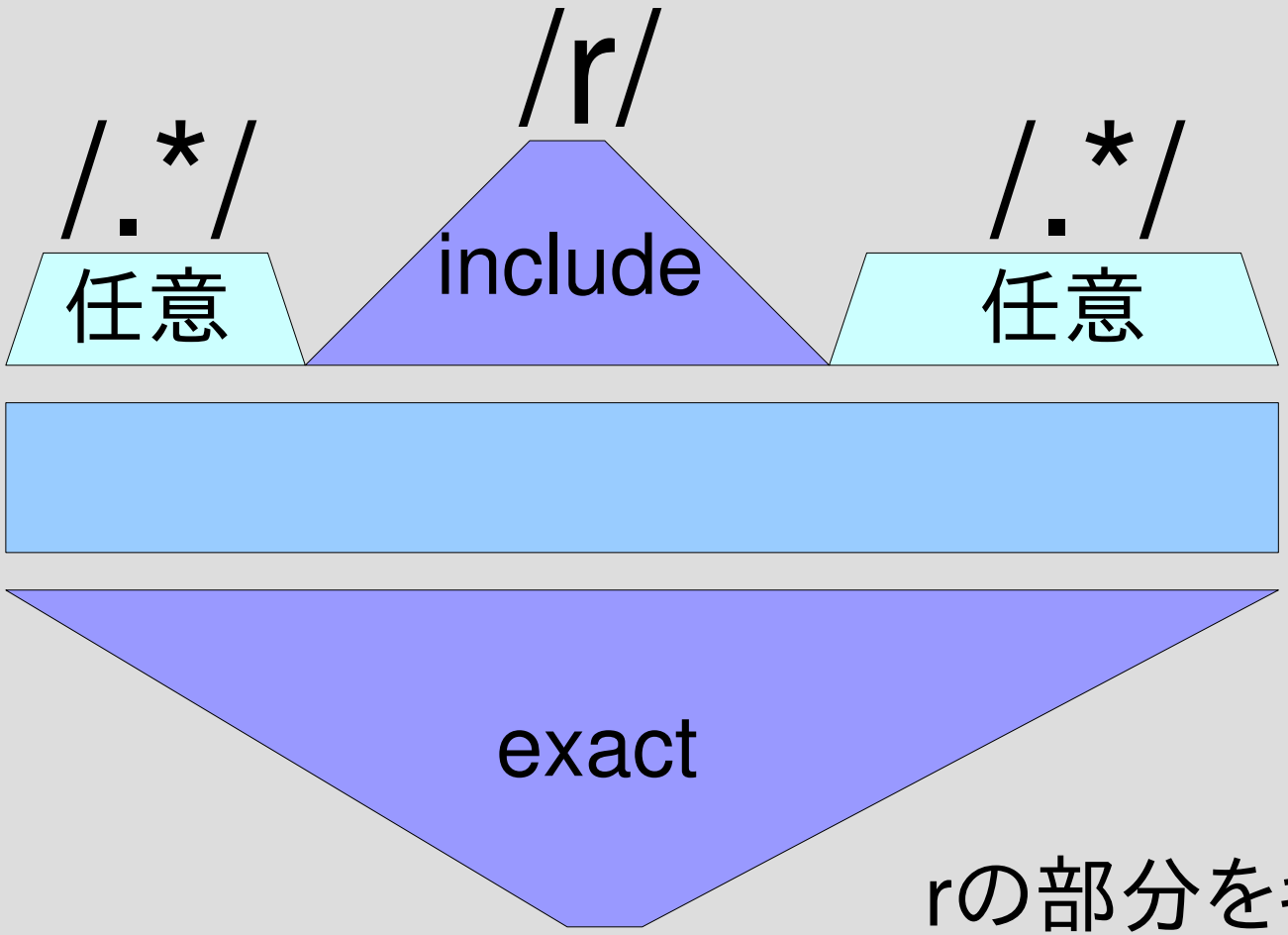
- p match_include([:cat, "n", "g"], "orange")
  #=> 3...5
- p match_include([:cat, "e", [:rep, "r"]],
  "berry")
  #=> 1...4
- p match_include([:anychar], "z")
  #=> 0...1

マッチの考え方



注: ここで /./ は
任意の一文字とする

match_include は
マッチした位置を返す



/.*(r).*/

rの部分をキャプチャ

r の位置を調べて返す

cap_exact による match_include の実装

```
def match_include(r, str)
  md = cap_exact(
    [:cat, [:rep, [:anychar]],
           [:capture, :match_inc, r],
           [:rep, [:anychar]]],
    str)
  return md[:match_inc] if md
  nil
end
```

match_include の動作の違い

- マッチする場所が 2箇所以上あるときに違う
- 講義で示した実装
  p match_include("a", "banana")　　#=> 1...2
- レポートの実装
  p match_include("a", "banana")　　#=> 5...6
- 講義のは左の方を見つける
  レポートのは右の方を見つける

ざっと見た結果

- 難しかった模様
- 部分文字列をいろいろと切り出して cap_exact というのが多かった
  - その場合、[:string_start] などの動作が変わる