

テキスト処理 第11回 (2008-07-01)

$r+?$ と $r\{m,\}$ の説明

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

- 以下を実装して解説せよ
 - $r+$?
 - $r\{m,\}$
- 実装したらユニットテストで確認すること
- ✕切 2008-07-01 12:00
- RENANDI
- 拡張子が txt なテキストファイル希望

各種繰り返し

	$0 \sim \infty$	$0 \sim 1$	$1 \sim \infty$
greedy	r^*	$r?$	r^+
lazy	$r^*?$	$r??$	$r+?$

	$m \sim n$	m	$m \sim \infty$
greedy	$r\{m,n\}$	$r\{m\}$	$r\{m,\}$
lazy	$r\{m,n\}?$	$r\{m\}?$	$r\{m,\}?$

/r+?/

- /r+/ の lazy 版
- 配列表現では [:plus_lazy, r]
- `rx_ends([:plus_lazy, "a"], "aaa", 0) #=> [1,2,3]`

plus_lazy の実装

```
def try_plus_lazy(re, str, pos, md, &b)
  try(re[1], str, pos, md) { |pos2, md2|
    try([:rep_lazy, re[1]], str, pos2, md2, &b)
  }
end
```

1回以上 = 1回 + 0回以上

ただし 0回以上のところで lazy 版を使う

/r{m,}/

- m回以上の r の繰り返し
 - 配列表現では [:moretimes, m, r]
 - greedy
-
- rx_ends([:moretimes, 3, "a"], "aa", 0) #=> []
 - rx_ends([:moretimes, 3, "a"], "aaa", 0) #=> [3]
 - rx_ends([:moretimes, 3, "a"], "aaaa", 0) #=> [4,3]
 - rx_ends([:moretimes, 3, "a"], "aaaaa", 0) #=> [5,4,3]

moretimes の実装

```
def try_moretimes(re, str, pos, md, &b)
  m = re[1]
  r = re[2]
  try(r, str, pos, md) { |pos2, md2|
    if pos < pos2
      try([:moretimes, m-1, r], str, pos2, md2, &b)
    end
  }
  yield pos, md if m <= 0
end
```

moretimes の実装方針

- try_times をベースにする
- n が無限大と考える
0 < n は常に真
- 無限再帰防止の検査を入れる
n をいくら減らしても 0 にならないので無限再帰の可能性が出てくる

try_times

```
def try_times(re, str, pos, md, &b)
  m = re[1]
  n = re[2]
  r = re[3]
  if 0 < n
    try(r, str, pos, md) { |pos2, md2|
      try([:times, m-1, n-1, r], str, pos2, md2, &b)
    }
  end
  yield pos, md if m <= 0
end
```

try_times から try_moretimes ^

```
def try_moretimes(re, str, pos, md, &b)
```

```
  m = re[1]
```

```
  n = re[2]
```

```
  r = re[3]
```

```
  if true
```

```
    try(r, str, pos, md) { |pos2, md2|
```

```
      try([:moretimes, m-1, n-1, r], str, pos2, md2, &b) if pos <
```

```
    }
```

```
  end
```

```
  yield pos, md if m <= 0
```

```
end
```

ざっと眺めた結果

- `try_plus_lazy` はだいたいできている
- `rep_lazy` を使わない実装もけっこうあった
- `try_moretimes` もだいたいできている
- 無限再帰防止は忘れがち
- `r{m,str.length}` とした実装もけっこうあった
 - `r` が空文字列にもマッチする場合に繰り返しが足りない場合がありうる