テキスト処理 第9回 (2007-07-03)
正規表現エンジン計算量レポート説明

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-2007/

レポート

- a が n 個並んでいる文字列に /a*aaa/ をマッチさせたときに try が呼び出される回数を n に対する関数として求めよ
- ∠切 2007-07-03 16:20
- HIPLUS
- 拡張子が txt なテキストファイル希望

/a*aaa/ =~ "a" * n

- ある特定の n に対し、try の呼び出し回数は以下で求められる
- count_try(
  [:cat, [:rep, [:lit, "a"]],
  [:cat, [:lit, "a"],
  [:cat, [:lit, "a"], [:lit, "a"]]]],
  "a" * n)

- 注意: /a*(a(aa))/ という構造とする

想定されるレポートの内容

- 求めた式
- その式が求まった理由
  - グラフから求めるのではなく、動作をたどって数えることを想定

ヒント

- わからなければ try の先頭に p exp とか p [pos, exp] とか入れて動作をたどる

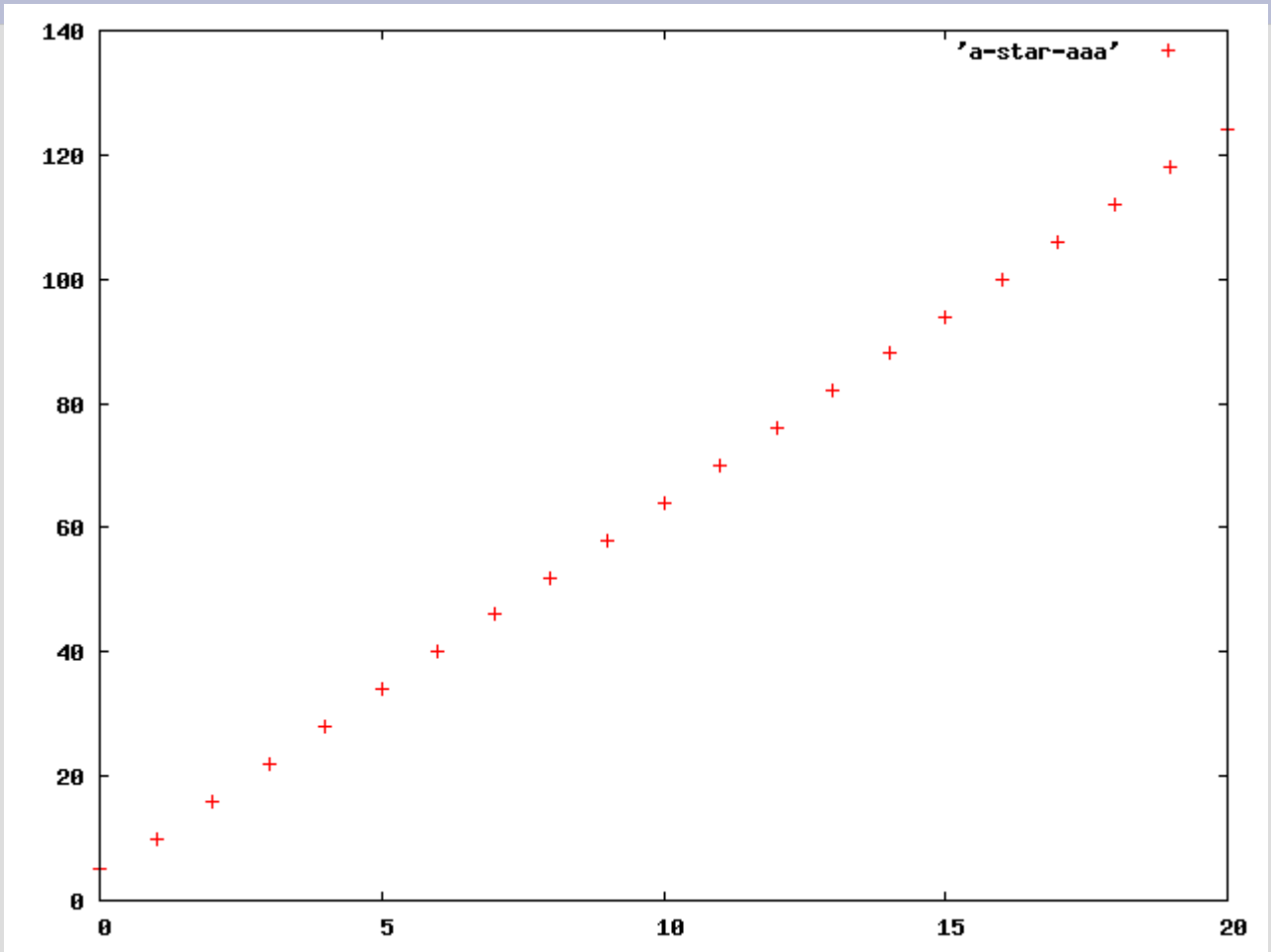
とりあえず実験

```
0.upto(20) {|n|
  m = count_try(
    [:cat, [:rep, [:lit, "a"]],
           [:cat, [:lit, "a"],
                  [:cat, [:lit, "a"], [:lit, "a"]]]],
    "a" * n)
  puts "#{n} #{m}"
}
```

とりあえず結果

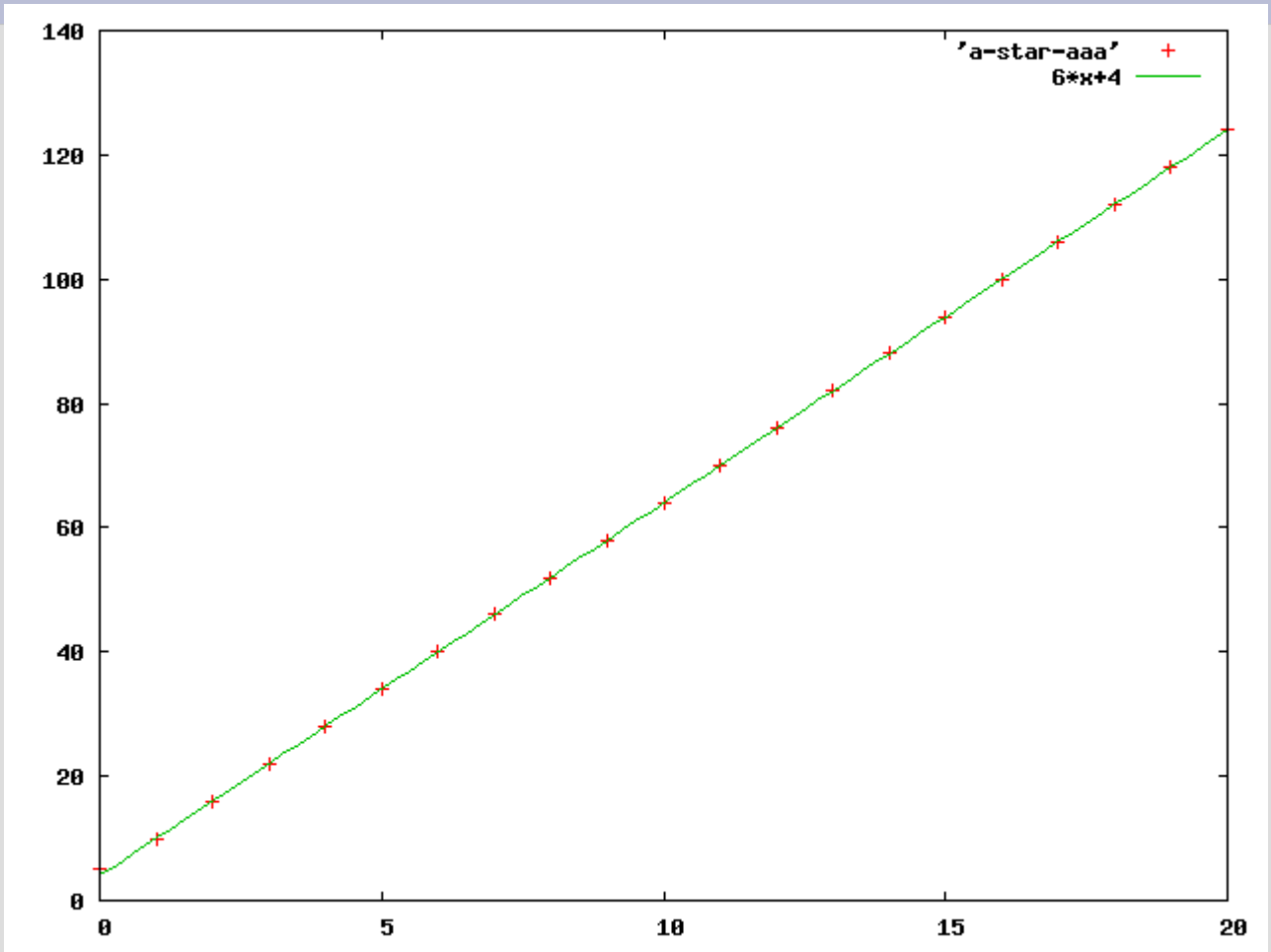
0 5
1 10
2 16
3 22
4 28
5 34
6 40
7 46
8 52
9 58
10 64
11 70
12 76
13 82
14 88
15 94
15 94
16 100
17 106
18 112
19 118
20 124

とりあえずプロット



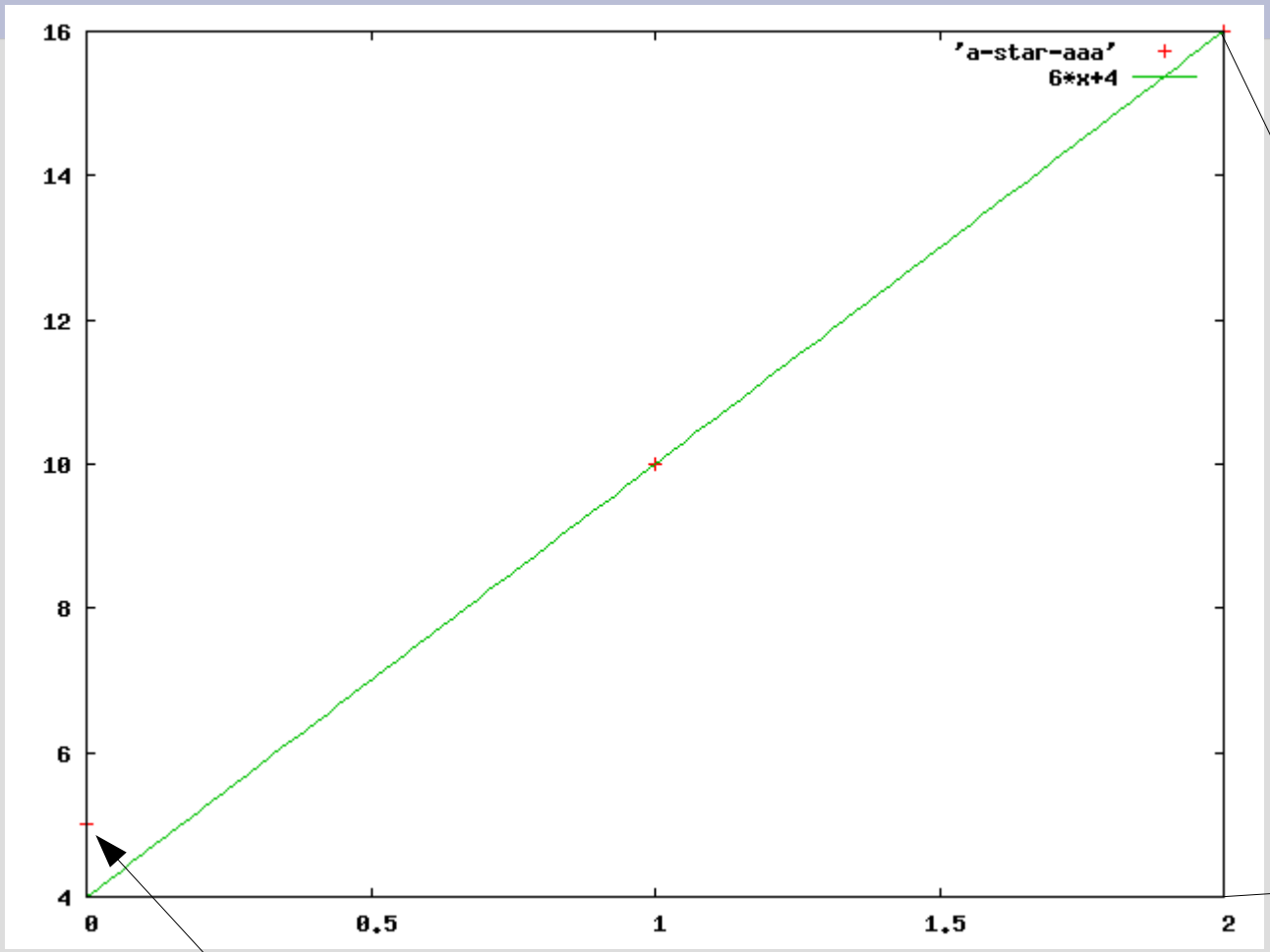
直線、か?

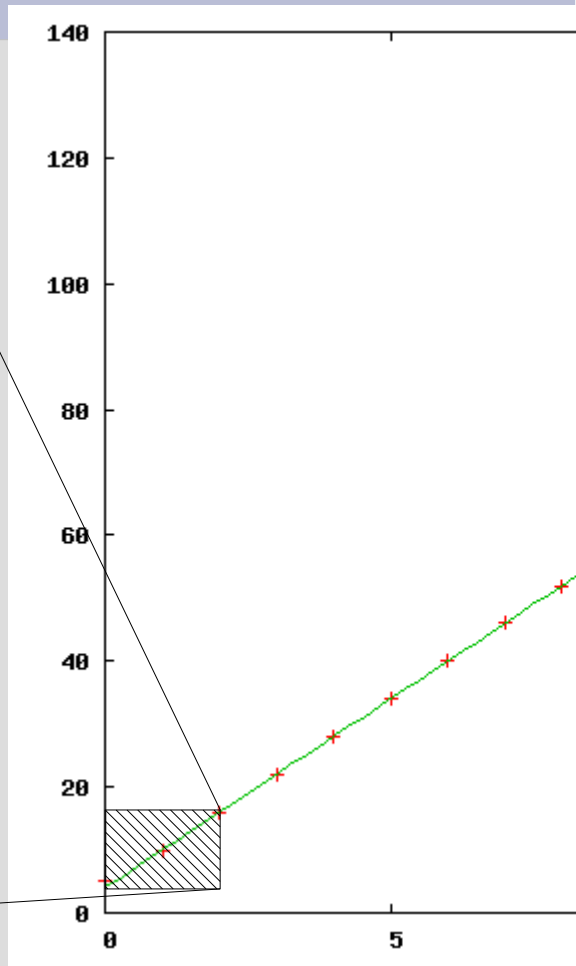
直線を引く



直線: 6n+4

原点付近を拡大





最初の点が直線に乗ってない

n個の a の並びに対する try の呼び出し回数

- n が 0 のとき 5
- n が 1以上のとき 6n+4

と、グラフからは読み取れる

try で pos と exp を表示

```
def try(exp, seq, pos, &block)
  p [pos, exp]
  $try_count += 1
  case exp[0]
```

count_try([...], "a"*1) = 10

[0, [:cat, [:rep, [:lit, "a"]], [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]]

[0, [:rep, [:lit, "a"]]]

[0, [:lit, "a"]]

[1, [:lit, "a"]]

[1, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]

[1, [:lit, "a"]]

[0, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]

[0, [:lit, "a"]]

[1, [:cat, [:lit, "a"], [:lit, "a"]]]

[1, [:lit, "a"]]

count_try([...], "a"*1) = 10

0 /a*aaa/	正規表現全体
0 /a*/	先頭の a* に挑戦
0 /a/	とりあえずひとつ a があって成功
1 /a/	次の a に挑戦 - 失敗
1 /aaa/	いちばん長く a* が伸びた 1 から aaa に挑戦
1 /a/	aaa の最初の a に挑戦 - 失敗
0 /aaa/	a* をひとつ短くして aaa にまた挑戦
0 /a/	aaaの最初の a に挑戦 - 成功
1 /aa/	残りの aa に挑戦
1 /a/	aa の最初の a に挑戦 - 失敗

count_try([...], "a"*1) = 10

0 /a*aaa/ ↓ 正規表現全体: 1

0 /a*/

0 /a/ a*のマッチ: 3=n+2

1 /a/

1 /aaa/

1 /a/ 位置1からの aaa のマッチ (失敗): 2

0 /aaa/

0 /a/

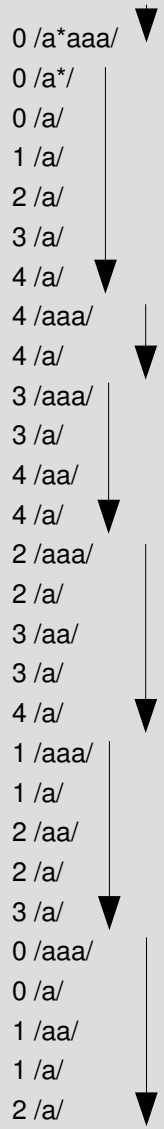
1 /aa/ 位置0からの aaa のマッチ (失敗): 4

1 /a/

count_try([...], "a"*4) = 28

[0, [:cat, [:rep, [:lit, "a"]], [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]]
[0, [:rep, [:lit, "a"]]]
[0, [:lit, "a"]]
[1, [:lit, "a"]]
[2, [:lit, "a"]]
[3, [:lit, "a"]]
[4, [:lit, "a"]]
[4, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]
[4, [:lit, "a"]]
[3, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]
[3, [:lit, "a"]]
[4, [:cat, [:lit, "a"], [:lit, "a"]]]
[4, [:lit, "a"]]
[2, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]
[2, [:lit, "a"]]
[3, [:cat, [:lit, "a"], [:lit, "a"]]]
[3, [:lit, "a"]]
[4, [:lit, "a"]]
[1, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]
[1, [:lit, "a"]]
[2, [:cat, [:lit, "a"], [:lit, "a"]]]
[2, [:lit, "a"]]
[3, [:lit, "a"]]
[0, [:cat, [:lit, "a"], [:cat, [:lit, "a"], [:lit, "a"]]]]
[0, [:lit, "a"]]
[1, [:cat, [:lit, "a"], [:lit, "a"]]]
[1, [:lit, "a"]]
[2, [:lit, "a"]]

count_try([...], "a"*4) = 28



正規表現全体: 1

a*のマッチ: 6=n+2

位置4からの aaa のマッチ (失敗): 2

位置3からの aaa のマッチ (失敗): 4

位置2からの aaa のマッチ (失敗): 5

位置1からの aaa のマッチ (成功): 5

位置0からの aaa のマッチ (成功): 5

count_try([...], "a"*1) = 10

0 /a*aaa/ ↓ 正規表現全体: 1

0 /a*/

0 /a/ a*のマッチ: 3=n+2

1 /a/

1 /aaa/

1 /a/ 位置1からの aaa のマッチ (失敗): 2

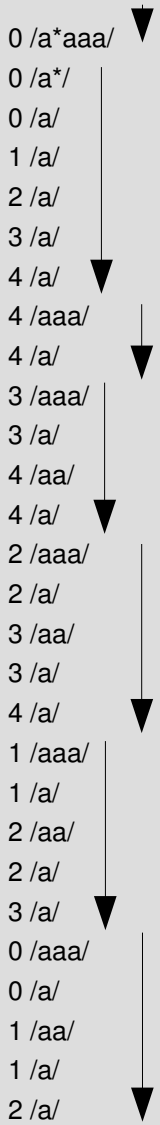
0 /aaa/

0 /a/

1 /aa/ 位置0からの aaa のマッチ (失敗): 4

1 /a/

count_try([...], "a"*4) = 28



正規表現全体: 1

a*のマッチ: 6=n+2

位置nからの aaa のマッチ (失敗): 2

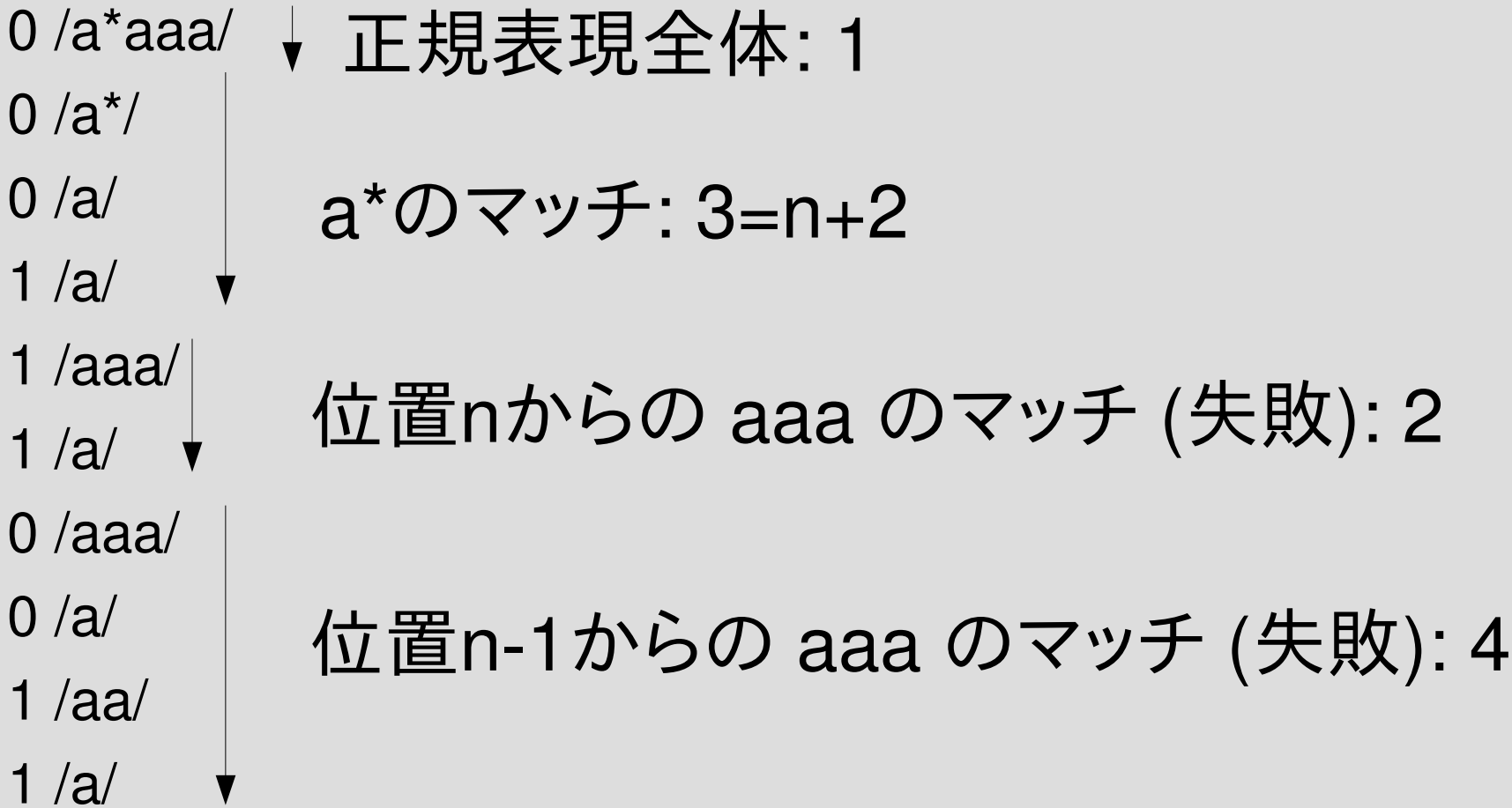
位置n-1からの aaa のマッチ (失敗): 4

位置n-2からの aaa のマッチ (失敗): 5

位置1からの aaa のマッチ (成功): 5

位置0からの aaa のマッチ (成功): 5

count_try([...], "a"*1) = 10



総計

- 正規表現全体: 1
- a* のマッチ: n+2
- i を n から 0 まで変えて以下を合計
  - i が n ならば 2
  - i が n-1 ならば 4
  - i が n-2 以下ならば 5

総計: n=1 のとき

- 正規表現全体: 1
- a* のマッチ: n+2=3
- i を n から 0 まで変えて以下を合計: 6
  - i が n ならば 2
  - i が n-1 ならば 4
  - i が n-2 以下にはならない

1+3+6=10

総計: n=4 のとき

- 正規表現全体: 1
- a* のマッチ: n+2=6
- i を n から 0 まで変えて以下を合計: 2+4+15=21
  - i が n ならば 2
  - i が n-1 ならば 4
  - i が n-2 以下ならば 5
    - n-2=2 で、i=0,1,2 の場合があるので 5*3 = 15

1+6+21=28

総計: 0 < n のとき

- 正規表現全体: 1
- a* のマッチ: n+2
- i を n から 0 まで変えて以下を合計: 2+4+5*(n-1)
  - i が n ならば 2
  - i が n-1 ならば 4
  - i が n-2 以下ならば 5
    - i=0,1,...,n-2 の場合があるので 5*(n-1)

$$1+n+2+2+4+5*(n-1)=6n+4$$

総計: n=0 のとき

- 正規表現全体: 1
- a* のマッチ: n+2=2
- i を n から 0 まで変えて以下を合計: 2
  - i が n ならば 2
  - i が n-1 にはならない
  - i が n-2 以下にはならない

1+2+2=5

6n+4に乗らないのは

i が n-1 のときが 4 だから

n個の a の並びに対する try の呼び出し回数

- n が 0 のとき 5
- n が 1以上のとき 6n+4

ざっと眺めた結果

- 6n+4 はだいたい求められた感じ
- n=0 の場合も扱っている人がそれなりにいる