

Count of Matches for a Highly Ambiguous Regular Expression

Mike French

Initial version: 2022-11-19

Latest version: 2022-12-20

Abstract

We evaluate the number of matches of the regular expression $(a?)^{\{m\}}(a^*)^{\{m\}}$, meaning m repetitions of each part, against the string 'a... ', containing m copies of the character 'a'.

The problem is a highly ambiguous exaggeration of the example given in [Cox].

We show the total count of matches is the dot product of two vectors taken from Pascal's Triangle.

A formula is given for the total count and values are calculated for $m=1..10$.

Definitions

Consider the match counts for each operator in the regular expression:

- Optional quantifier *zero or one* '?' matches 0 or 1 characters.
The counts for the first half of the expression ' $?^{\{m\}}$ ' form a sequence of m binary digits
- Kleene Star quantifier *zero or more* '*' matches $0..m$ characters.
The counts for the second half ' $*^{\{m\}}$ ' form a sequence of m numbers in the range $0..m$.

Count the total number of ways to get a specific partial sum of matches $k=0..m$ for each half of the expression. Assemble these counts into two vectors of $m+1$ values over index $k=0..m$:

- $S_{?m}[k]$ ways for ' $?^{\{m\}}$ ' to match k characters.
- $S_{*m}[k]$ ways for ' $*^{\{m\}}$ ' to match k characters

For a successful match, the two counts for each half of the expression must add up to m . So if the second half matches k , the first half must have matched $m-k$.

Then the total count is the pairwise multiplication of the S vectors:

$$\text{Total count : } S_m = \sum_{k=0..m} S_{?m}[m-k] \times S_{*m}[k]$$

? Quantifiers

The count value $S_{?m}[k]$ is:

- The number of ways to get m zero-or-one matches accepting a total of k characters.
- The count of m -digit binary numbers that have k bits set (1s).
- The number of ways of choosing k from m , which is the binomial coefficient nC_k , with $n=m$:

$$S_{?m}[k] = {}^mC_k$$

The table of binomial coefficients is just Pascal's Triangle with the recurrence relation:

$${}^nC_k = ({}^{n-1}C_{k-1}) + ({}^{n-1}C_k)$$

The vector of counts $S_{?m}[k]$ is mC_k for $k=0..m$, which is just a diagonal vector in Pascal's Triangle.

The diagonal vector is symmetric because: ${}^nC_k = {}^nC_{n-k}$ therefore $S_{?m}[k] = S_{?m}[m-k]$

which means we can invert the $S_{?m}$ vector index and justify the dot product formulation:

$$\text{Total count: } S_n = \sum_{k=0..m} S_{?m}[k] \times S_{*m}[k] = S_{?m} \bullet S_{*m}$$

* Quantifiers

The count value $S_{*m}[k]$ is:

- The number of ways to get m zero-or-more matches accepting a total of k characters.
- *Sum of digits* problem: the ways m numbers each in the range $0..m$ can have sum k .

Construct a recurrence relation for the sum of digits problem. Each set of $m-1$ numbers with sum s in the range $0 \leq s \leq k$ is uniquely made up to sum k by adding the m^{th} number $k-s$:

$$S_{*m}[k] = \sum_{k=0..m} S_{*m-1}[k] = S_{*m}[k-1] + S_{*m-1}[k]$$

Each entry (m,k) is the sum of values in the row above, up to and including the same column k , and hence also the sum of the two terms to the left $(m, k-1)$ and above $(m-1, k)$.

The recurrence is grounded at 1 on the left and at the top:

$$\forall_{k=0..m} S_{*1}[k] = 1, \quad \forall_{m>0} S_{*m}[0] = 1$$

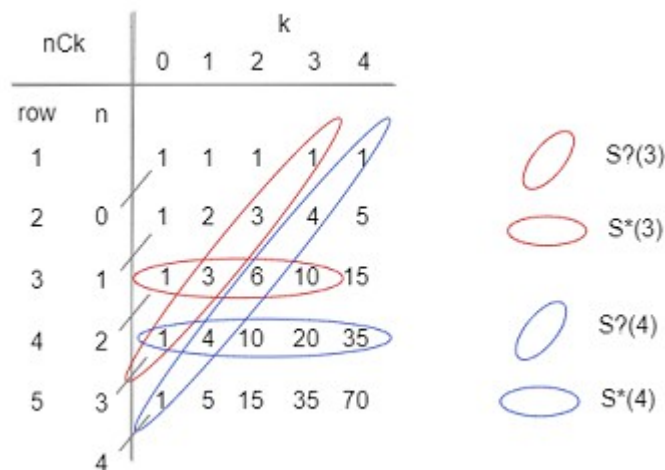
This is just Pascal's Triangle accessed by the 1-based row number m , rather than n , the 0-based diagonal. The indices are linked by $n=m+k-1$, so the count is the binomial coefficient:

$$S_{*m}[k] = (m+k-1) C k$$

Final Formula

$$\text{Total count: } S_m = S_{?m} \cdot S_{*m} = \sum_{k=0..m} m C k \times (m+k-1) C k$$

Vectors in Pascal's Triangle



$$\begin{aligned}
 S(1) &= [1,1] \cdot [1,1] = 1+1 = 2 \\
 S(2) &= [1,2,1] \cdot [1,2,3] = 1+4+3 = 8 \\
 S(3) &= [1,3,3,1] \cdot [1,3,6,10] = 1+9+18+10 = 38 \\
 S(4) &= [1,4,6,4,1] \cdot [1,4,10,20,35] = 1+16+60+80+35 = 192 \\
 S(5) &= [1,5,10,10,5,1] \cdot [1,5,15,35,70,126] = 1+25+150+350+350+126 = 1,002 \\
 S(6) &= [1,6,15,20,15,6,1] \cdot [1,6,21,56,126,252,462] = 1+36+315+1120+1890+1512+462 = 5,336
 \end{aligned}$$

n	1	2	3	4	5	6	7	8	9	10
S_n	2	8	38	192	1,002	5,336	28,814	157,184	864,146	4,780,008

References

[Cox] "Regular Expression Matching Can Be Simple And Fast", Russ Cox, January 2007 [\[web\]](#).