Motif Statistics

Pierre Nicodème

LIX, École polytechnique

C.N.R.S.

Joint work with

Bruno Salvy and *Philippe Flajolet*

Inria - Rocquencourt

Regular expressions

$$R = (a \cdot b \cdot a + (c^4 \cdot e)^\star \cdot b \cdot b)^\star$$

Operators

- $+$ Union
- $\cdot$ Concatenation
- $\star$ Star-operator $(A^\star = \epsilon + A + A^2 + A^3 + \dots)$

Aim & Result

R given regular expression.

X_n number of occurrences in a text of length n.

$$\text{Aim:} \qquad F(z,u) = \sum_{n,k} \Pr(X_n = k) u^k z^n.$$

Theorem. With or without counting overlap, both in the Bernoulli and Markov model,

($i.$) $F(z,u)$ is rational and can be computed explicitly

$$(ii.) \qquad \begin{cases} \mathrm{E}(X_n) & = \mu n + c_1 + O(A^n), \\ \mathrm{Var}(X_n) & = \sigma^2 n + c_2 + O(A^n). \end{cases}$$

($iii.$) Limit Gaussian law:

$$\Pr\left(\frac{X_n - \mu n}{\sigma\sqrt{n}}\right) \to \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt.$$

Generating functions on languages

$\mathcal{L} \in \Sigma^\star \qquad (\Sigma = \{l_1, l_1, \ldots, l_k\}$ alphabet$)$

Counting generating function

$F(z) = \sum_{\alpha \in \mathcal{L}} z^{|\alpha|} = \sum f_n z^n \qquad (|\alpha|$ taille de $\alpha)$

Multivariate generating function

$M(l_1, l_2, \ldots, l_p) = \sum_{\alpha \in \mathcal{L}} \mathrm{commute}(\alpha)$

Examples

$\Sigma = \{a, b\} \quad (\epsilon$ empty word$)$

$\mathcal{L} = \{\epsilon, aa, ab, ba, aaab\}$

$$\Rightarrow \begin{cases} F(z) = 1 + 3z^2 + z^4 \\ M(a,b) = 1 + a^2 + 2ab + a^3 b \end{cases}$$

$$\mathcal{L} = \{\epsilon, aab, aabaab, \ldots, (aab)^n, \ldots\} \qquad \Rightarrow \begin{cases} F(z) = \dfrac{1}{1 - z^3} \\ M(a,b) = \dfrac{1}{1 - a^2 b} \end{cases}$$

Weighted generating function

Bernoulli model, ω_i proba. of letter l_i,

π_α probability of word α = product of proba. of the letters of the word

Univariate generating function $F_\omega(z) = \sum_{\alpha \in \mathcal{L}} \pi_\alpha z^{|\alpha|} = \sum \pi_n z^n$

π_n proba. that a word of size n belongs to $\mathcal{L}$

Multivariate generating function

$M_\omega(l_1, l_2, \ldots, l_p) = \sum_{\alpha \in \mathcal{L}} \pi_\alpha \times \text{commute}(\alpha)$

Examples

$$\begin{aligned} \Sigma &= \{a, b\} \quad \omega_a = 1/3,\ \omega_b = 2/3 \\ \mathcal{L} &= \{\epsilon, aa, ab, ba, aaab\} \end{aligned} \qquad \Rightarrow \begin{cases} F_\omega(z) = 1 + \frac{5}{9} z^2 + \frac{2}{81} z^4 \\ M_\omega(a,b) = 1 + \frac{1}{9} a^2 + \frac{4}{9} ab + \frac{2}{81} a^3 b \end{cases}$$

Remark

$M(a,b) = 1 + a^2 + 2ab + a^3 b \qquad M(\omega_a z, \omega_b z) = F_\omega(z)$

Combinatorial Constructions $\Rightarrow$ Generating functions

Product

If $A_1.A_2 \dots A_j$ non ambiguous,
$F_{A_1.A_2 \dots A_j}(z) = F_{A_1}(l_1, \dots, l_k) \dots F_{A_j}(l_1, \dots, l_k)$

Union

If A and B disjoint, $F_{A \cup B}(l_1, \dots, l_k) = F_A(l_1, \dots, l_k) + F_B(l_1, \dots, l_k)$

Kleene $\star$ operator

If no ambiguity, $F_{A^*}(l_1, \dots, l_k) = \dfrac{1}{1 - F_A(l_1, \dots, l_k)}$

Counter-example

$$A_1 = A_2 = \{a, aa\}, \quad \Sigma = \{a\}$$

$$\Rightarrow \begin{cases} A_1 A_2 = \{aa, aaa, aaaa\} \quad \Longrightarrow \quad \left| \begin{array}{l} F_{A_1 A_2}(a) = a^2 + a^3 + a^4 \\ \neq F_{A_1}(a) F_{A_2}(a) = a^2 + 2a^3 + a^4 \end{array} \right. \\ A_1 \bigcup A_2 = \{a, aa\} \quad \Longrightarrow \quad F_{A_1 \bigcup A_2}(a) = a + a^2 \neq F_{A_1}(a) + F_{A_2}(a) = 2a + 2a^2 \end{cases}$$

The Right Rational Language

R regular expression over Σ

Key: find an algorithmic way (automaton) to insert in each word of $\Sigma^\star$ a mark (empty size fake letter) (m) after each occurrence of R.

Example: $R = aba$

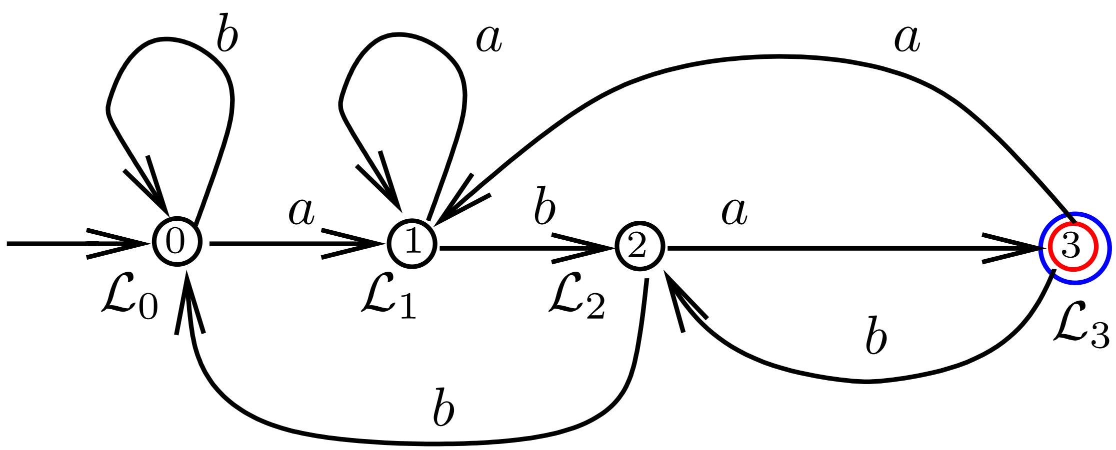
$aaaabambamaaabamaa$ (overlap)

$aaaabamba \quad aaabamaa$ (non-overlap).

Automaton Recognizing $\Sigma^* R$

$$\Sigma = \{a, b\} \quad R = aba, \quad E = \Sigma^* R = \Sigma^* aba$$

$$aabb\,aba \bullet bbabb\,aba \bullet aa\,aba \bullet bbbb$$



Chomsky-Schützenberger

$$\mathcal{L}_0 = a\mathcal{L}_1 + b\mathcal{L}_0, \qquad \mathcal{L}_1 = b\mathcal{L}_2 + a\mathcal{L}_1,$$

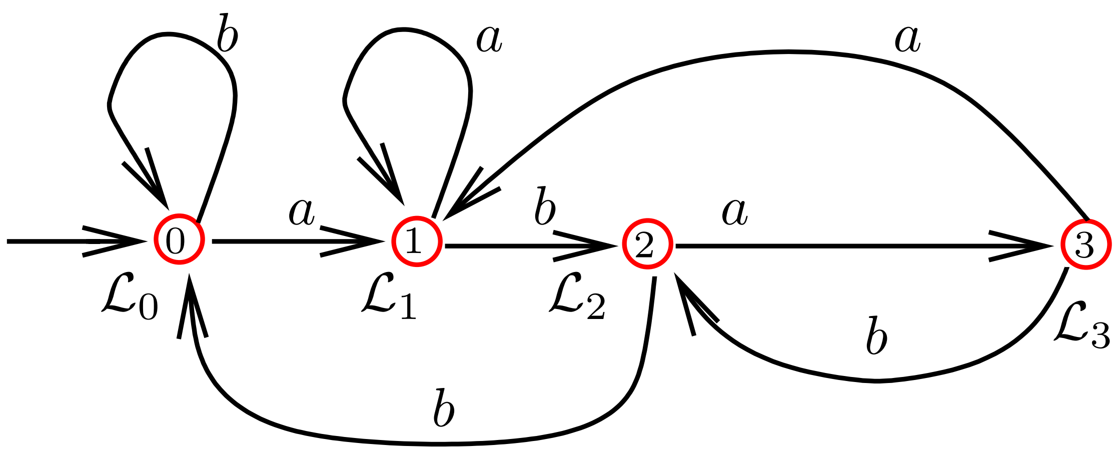
$$\mathcal{L}_2 = a\mathcal{L}_3 + b\mathcal{L}_0, \qquad \mathcal{L}_3 = a\mathcal{L}_1 + b\mathcal{L}_2 + \epsilon,$$

$$\begin{cases} L_0 = aL_1 + bL_0, & L_1 = bL_2 + aL_1, \\ L_2 = aL_3 + bL_0, & L_3 = aL_1 + bL_2 + 1. \end{cases} \implies \quad L_0(a, b) = \frac{a^2 b}{1 - a - b}$$

Automaton Recognizing $\Sigma^* R$

$$\Sigma = \{a, b\} \quad R = aba, \quad E = \Sigma^* R = \Sigma^* aba$$

$$aabb\,aba\bullet bbabb\,aba\bullet aa\,aba\bullet bbbb$$



Chomsky-Schützenberger

$$\mathcal{L}_0 = a\mathcal{L}_1 + b\mathcal{L}_0 + \epsilon, \qquad \mathcal{L}_1 = b\mathcal{L}_2 + a\mathcal{L}_1 + \epsilon,$$

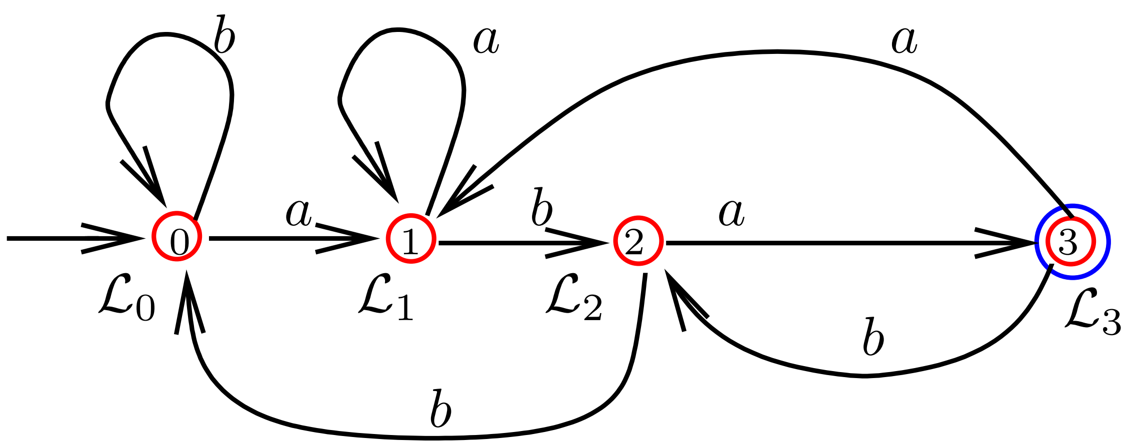
$$\mathcal{L}_2 = a\mathcal{L}_3 + b\mathcal{L}_0 + \epsilon, \qquad \mathcal{L}_3 = a\mathcal{L}_1 + b\mathcal{L}_2 + \epsilon,$$

$$\left\{ \begin{array}{ll} L_0 = aL_1 + bL_0 + 1, & L_1 = bL_2 + aL_1 + 1, \\ L_2 = aL_3 + bL_0 + 1, & L_3 = aL_1 + bL_2 + 1. \end{array} \right. \implies L_0(a,b) = \frac{1}{1-(a+b)}$$

Automaton Recognizing $\Sigma^* R$

$$\Sigma = \{a, b\} \quad R = aba, \quad E = \Sigma^* R = \Sigma^* aba$$

$$aabb\,aba \bullet bbabb\,aba \bullet aa\,aba \bullet bbbb$$



$$\begin{cases} L_0 = aL_1 + bL_0 + 1, & L_1 = bL_2 + aL_1 + 1, \\ L_2 = amL_3 + bL_0 + 1, & L_3 = aL_1 + bL_2 + 1. \end{cases}$$

$$\Longrightarrow \quad L_0 = L(a, b, m) = \frac{1 + ab(1 - m)}{1 - a - b + ab(1 - m) - ab^2(1 - m)}$$

Generating functions counting matches with aba

$$L(a,b,m) = \frac{1+ab(1-m)}{1-a-b+ab(1-m)-ab^2(1-m)}$$

$$F(z,u) = L(\pi_a z, \pi_b z, u) = \frac{1+\pi_a\pi_b z^2(1-u)}{1-z+\pi_a\pi_b z^2(1-u)-\pi_a\pi_b^2 z^3(1-u)}$$

$$= \frac{1}{1-z+\pi_{aba} z^{|aba|}\dfrac{1-u}{u+(1-u)C_{aba}(z)}}$$

$\{\epsilon, ba\}$ autocorrelation set of aba

$C_{aba}(z) = 1 + \pi_a\pi_b z^2$ autocorrelation polynomial of the word aba

From Regular Expression to NFA by Berry-Sethy

$E = (a + b)^* aba$

1) mark letters occurrences $E' = (a_1 + b_1)^* a_2 b_2 a_3$

2) use the constructors $\mathrm{first}, \mathrm{last}, \mathrm{follow}$

$\mathrm{first}(R) = \{a_1, b_1, a_2\}$

$\mathrm{last}(R) = \{a_3\}$

$\mathrm{follow}(R, b_1) = \{a_1, b_1, a_2\}$

3) automaton:

marked letters $\rightarrow$ state,

suppress indices $\rightarrow$ transitions

$\delta(b_1, a) = \{a_1, a_2\}, \quad \delta(b_1, b) = \{b_1\}$

Berry-Sethy Algorithm

recursive definition of first, last, follow and nullable

nullable$(R) = \textit{true}$ if $\epsilon \in$ language of R

first$(R_1 R_2) =$

$$\begin{cases} \text{first}(R_1) \cup \text{first}(R_2) \quad \text{if} \quad \text{nullable}(R_1), \\ \text{first}(R_1) \quad \text{elsewhere} \end{cases}$$

follow$(R_1 R_2, x) =$

$$\begin{cases} \text{follow}(R_2, x) \quad \text{if} \quad x \in R_2, \\ \text{follow}(R_1, x) \cup \text{first}(R_2) \quad \text{if} \quad x \in \text{last}(R_1) \\ \text{follow}(R_1, x) \quad \text{elsewhere} \end{cases}$$

follow$(R^*, x) =$

$$\begin{cases} \text{follow}(R, x) \cup \text{first}(R) \quad \text{if} \quad x \in \text{last}(R), \\ \text{follow}(R, x) \quad \text{elsewhere} \end{cases}$$

Technical condition $\Rightarrow$ quadratic complexity

The algorithmic chain

Input: regular expression R

1. Berry-Sethy $\mapsto$ NFA for $\Sigma^* R$
2. Determinisation $\mapsto$ DFA for $\Sigma^* R$
3. Marking $\mapsto$ marked DFA for $\Sigma^* R$
4. Chomsky-Schützenberger $\mapsto$ $F(z,u)$,

$$F(z,u) = \sum p_{n,k} u^k z^n,$$

$p_{n,k}$: probability that a word of size n contains k occurrences of R.

Exploiting the Output

$$F(z,u) \in \mathbb{Q}(z,u) \Rightarrow \begin{cases} G(z) = \sum \mathrm{E}(X_n) z^n \in \mathbb{Q}(z), \\ H(z) = \sum M_2(X_n) z^n \in \mathbb{Q}(z), \\ N(z) = \sum \Pr(X_n \geq 1) z^n \in \mathbb{Q}(z). \end{cases}$$

– Fast extraction of coefficients: nth coefficient in $O(\log n)$ operations [implemented in gfun].

– Exponentially good asymptotics in constant time.

Proof of the Gaussian Law

$$
\begin{aligned}
L_0(z,u) &= z\pi_a L_1 + z\pi_b L_0 + 1, \\
L_1 &= z\pi_b L_2 + z\pi_a L_1 + 1, \\
L_2 &= z\pi_a u L_3 + z\pi_b L_0 + 1 \\
L_3 &= z\pi_a L_1 + z\pi_b L_2 + 1
\end{aligned}
$$

$$
L = \begin{pmatrix} L_0 \\ \vdots \\ L_n \end{pmatrix} = z\,\mathbf{T}(u)L + \mathbf{1}
$$

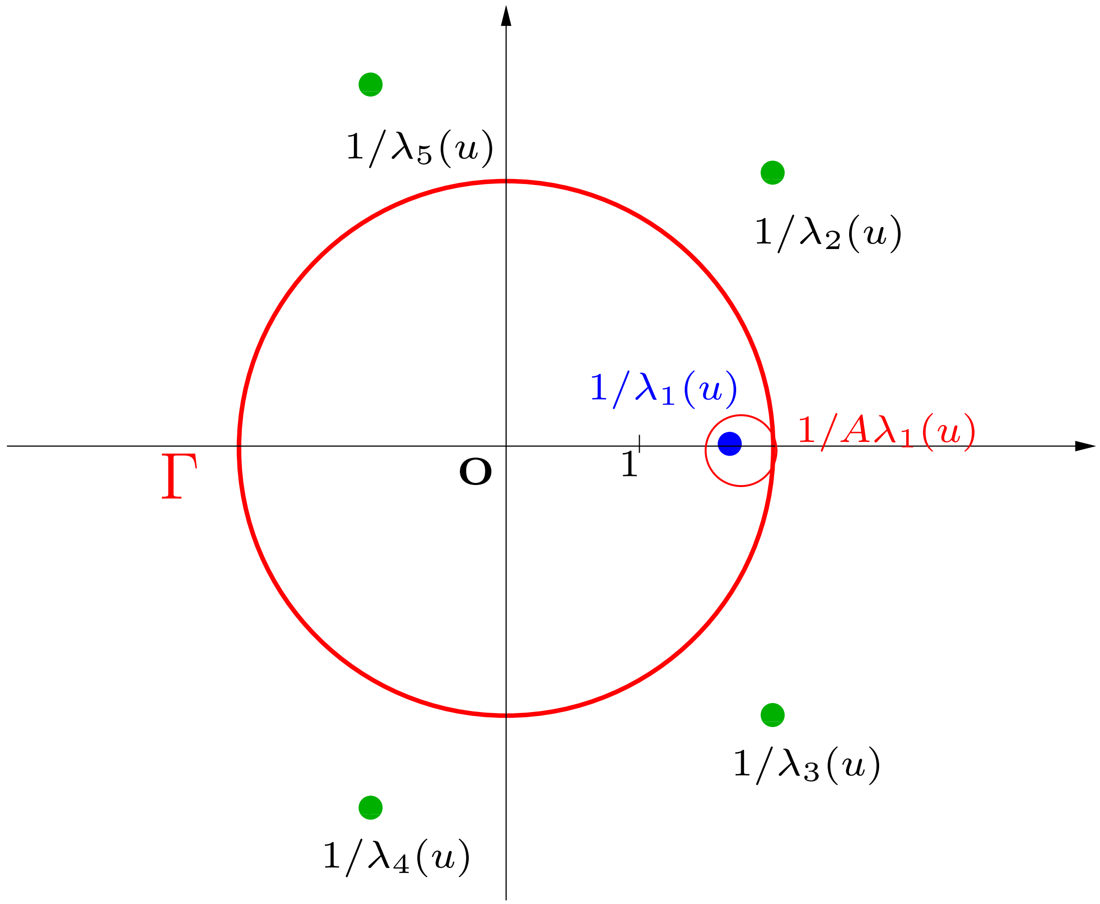
$\mathbf{T}(u)$ positive $n \times n$ matrix

$$
L_0(z,u) = \frac{P(z,u)}{Q(z,u)} = \frac{P(z,u)}{(1 - z\lambda_1(u)) \cdots (1 - z\lambda_n(u))}
$$

$$
1/|\lambda_1| \leq 1/|\lambda_2| \leq \ldots
$$

Perron-Frobenius: $\lambda_1(u)$ unique, real, positive.

Uniform Separation Property



$$\pi_n(u) = [z^n]F(z,u) = \frac{1}{2i\pi}\oint_\Gamma \frac{dz}{z^{n+1}}F(z,u),$$

$$= \frac{1}{2i\pi}\oint_\Gamma \frac{c(u)}{z^{n+1}(1-\lambda_1(u)z)} + \frac{1}{z^{n+1}}g(z,u)\,dz = c(u)\lambda_1(u)^n(1+O(A^n)).$$

Hwang's quasi-power theorem → limiting Gaussian distribution.

Application : Prosite Motifs

```
AC   PS00723;
DE   Polyprenyl synthetases signature 1.
...
PA   [LIVM](2)-x-D-D-x(2,4)-D-x(4)-R-R-[GH].

...
DR   P14324, FPPS_HUMAN, T; ... P49353, FPPS_MAIZE, T;
DR   P08524, FPPS_YEAST, T; ... P08836, FPPS_CHICK, P;
...
```

Biological pertinence of motifs

with respect to a target genome

More generally: statistics of the number of occurrences of a regular expression in a random text.

Maple Demo $R = aba$ - http://algo.inria.fr/libraries

```
> with(regexpcount): with(combstruct): with(gfun):readlib(equivalent):
> G:={R=Prod(a,b,a),a=Atom,b=Atom}:
> Auto:=regexptomatchesgram(G,S,[[R,m,'overlap']]);
```

$$Auto := \{w3 = \mathrm{Union}(\mathrm{Prod}(a, m, w2), \mathrm{E}, \mathrm{Prod}(b, S)), a = Atom, b = Atom, w4 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, w3)), S = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, S)), w2 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, w3)), m = \mathrm{E}\}$$

```
> Fcount:=subs(gfsolve(Auto,unlabelled,z,[[u,m]]),S(z,u));
```

$$Fcount := -\frac{-z^2 - 1 + z^2 u}{z^3 u - z^3 + 1 - 2z + z^2 - z^2 u}$$

```
> FBernUnif:=subs(z=z/2,Fcount);
```

$$FBernUnif := -\frac{-\frac{1}{4} z^2 - 1 + \frac{1}{4} z^2 u}{\frac{1}{8} z^3 u - \frac{1}{8} z^3 + 1 - z + \frac{1}{4} z^2 - \frac{1}{4} z^2 u}$$

```
> Fexpect:=normal(subs(u=1,diff(FBernUnif,u)));
```

$$Fexpect := \frac{1}{8} \frac{z^3}{(-1+z)^2}$$

```
> expect:=convert(equivalent(Fexpect,z,n,2),polynom);
```

$$expect := \frac{1}{8} n - \frac{1}{4}$$

```
> Fmom2:=normal(subs(u=1,diff(u*diff(FBernUnif,u),u)));
```

$$Fmom2 := \frac{1}{32} \frac{z^3 (-4 + 4z + z^3 - 2z^2)}{(-1+z)^3}$$

```
> mom2:=convert(equivalent(Fmom2,z,n,3),polynom);
```

$$mom2 := \frac{1}{64} n^2 + \frac{3}{64} n - \frac{3}{16}$$

```
> std:=sqrt(mom2-expect^2);
```

$$std := \frac{1}{8} \sqrt{7n - 16}$$

Maple Demo $R = ab^+a$ - http://algo.inria.fr/libraries

```
> with(regexpcount): with(combstruct): with(gfun):readlib(equivalent):
> G:={R=Prod(a,b,Sequence(b),a),a=Atom,b=Atom}:
> Auto:=regexptomatchesgram(G,S,[[R,m,'overlap']]);
```

$$Auto := \{a = Atom, b = Atom, w3 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(b, w3), \mathrm{Prod}(a, m, w2)), w4 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, w3)), S = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, S)), w2 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, w3)), m = \mathrm{E}\}$$

```
> Fcount:=subs(gfsolve(Auto,unlabelled,z,[[u,m]]),S(z,u));
```

$$Fcount := -\frac{-z^2 - 1 + z + z^2 u}{z^3 u + 1 - 3z + 3z^2 - z^3 - z^2 u}$$

```
> FBernUnif:=subs(z=z/2,Fcount);
```

$$FBernUnif := -\frac{-\frac{1}{4} z^2 - 1 + \frac{1}{2} z + \frac{1}{4} z^2 u}{\frac{1}{8} z^3 u + 1 - \frac{3}{2} z + \frac{3}{4} z^2 - \frac{1}{8} z^3 - \frac{1}{4} z^2 u}$$

```
> Fexpect:=normal(subs(u=1,diff(FBernUnif,u)));
```

$$Fexpect := -\frac{1}{4} \frac{z^3}{(z-1)(2 - 3z + z^2)}$$

```
> expect:=convert(equivalent(Fexpect,z,n,2),polynom);
```

$$expect := \frac{1}{4} n - \frac{3}{4}$$

```
> Fmom2:=normal(subs(u=1,diff(u*diff(FBernUnif,u),u)));
```

$$Fmom2 := \frac{1}{8} \frac{z^3 (z^2 - 2z + 2)}{(z-1)^2 (2 - 3z + z^2)}$$

```
> mom2:=convert(equivalent(Fmom2,z,n,3),polynom);
```

$$mom2 := \frac{1}{16} n^2 - \frac{5}{16} n + \frac{5}{8}$$

```
> std:=sqrt(mom2-expect^2);
```

$$std := \frac{1}{4} \sqrt{n+1}$$

Maple Demo $R = ab^+a$ - `http://algo.inria.fr/libraries`

```
> with(regexpcount): with(combstruct): with(gfun):readlib(equivalent):
> G:={R=Prod(a,b,Sequence(b),a),a=Atom,b=Atom}:
> Auto:=regexptomatchesgram(G,S,[[R,m,'renewal']]);
```

$$Auto := \{w2 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, S)), w3 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, m, w2), \mathrm{Prod}(b, w3)), a = Atom, b = Atom, w4 = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, w3)), S = \mathrm{Union}(\mathrm{E}, \mathrm{Prod}(a, w4), \mathrm{Prod}(b, S)), m = \mathrm{E}\}$$

```
> Fcount:=subs(gfsolve(Auto,unlabelled,z,[[u,m]]),S(z,u));
```

$$Fcount := -\frac{1 - z + z^2}{z^3\, u - 1 + 3\, z - 3\, z^2 + z^3}$$

```
> FBernUnif:=subs(z=z/2,Fcount);
```

$$FBernUnif := -\frac{1 - \frac{1}{2}\, z + \frac{1}{4}\, z^2}{\frac{1}{8}\, z^3\, u - 1 + \frac{3}{2}\, z - \frac{3}{4}\, z^2 + \frac{1}{8}\, z^3}$$

```
> Fexpect:=normal(subs(u=1,diff(FBernUnif,u)));
```

$$Fexpect := \frac{1}{2}\, \frac{z^3}{(-1 + z)\,(z^3 - 4 + 6\, z - 3\, z^2)}$$

```
> expect:=convert(equivalent(Fexpect,z,n,2),polynom);
```

$$expect := \frac{1}{6}\, n - \frac{1}{3}$$

```
> Fmom2:=normal(subs(u=1,diff(u*diff(FBernUnif,u),u)));
```

$$Fmom2 := -\frac{1}{2}\, \frac{z^3\,(4 - 6\, z + 3\, z^2)}{(-1 + z)\,(z^3 - 4 + 6\, z - 3\, z^2)^2}$$

```
> mom2:=convert(equivalent(Fmom2,z,n,3),polynom);
```

$$mom2 := \frac{1}{36}\, n^2 - \frac{1}{12}\, n + \frac{5}{27}$$

```
> std:=sqrt(mom2-expect^2);
```

$$std := \frac{1}{18}\sqrt{9\, n + 24}$$