A alfabeto , $|A| = m$

p pattern, $p \in A^*$, $|p| = k$
$S(t)$ f.g. delle parole su A che evitano p (rispetto alla lunghezza delle parole)
$[t^n]S(t) = S_n$ numero di parole del linguaggio che evitano p e hanno lunghezza n.

Guibas e Oblyzko (1981)

$S(t) = \frac{C(t)}{t^k+(1-mt)c(t)}$
dove c(t) è il polinomio di autocorrelazione di p.

$A = a, b, \quad m = 2$
$p = aabbaa \quad k = 6$

aabbaa	$CodaPattern$	$esito$
aabba		$1\ C_0$
aabba	a	$0\ C_1$
aabb	aa	$0\ C_2$
aab	baa	$0\ C_3$
aa	$bbaa$	$1\ C_4$
a	$abbaa$	$1\ C_5$

$c(t) = \sum_{i=0}^{k-1} c_i t^i$
i corrisponde alla lunghezza della coda del pattern.
$c(t) = \sum_{i=0}^{k-1} c_i t^i = c_0 + c_4 t^4 + c_5 t^5$
$S(t) = \frac{1+t^4+t^5}{t^6+(1-2t)(1+t^4+t^5)}$

Notazione di Iversion

Se si parte da un pattern $p = p_1 p_2 \ldots p_k$
$C_i = [p_1 p_2 \ldots p_{k-1} = p_{i+1} p_{i+2} \ldots p_k]$
Nel nostro caso $p = aabbaa$
$C_0 = [aabbaa = aabbaa] = 1$
$C_1 = [aabba = abbaa] = 0$ non vale
$C_2 = [aabb = bbaa] = 0$
$C_3 = [aab = baa] = 0$

$C_4 = [aa = aa] = 1$
$C_5 = [a = a] = 1$

$p = p_1p_2 \dots p_k$
$C_i = [p_1p_2 \dots p_{k-1} = p_{i+1}p_{i+2} \dots p_k]$
S è l insieme delle parole che evitano p.
T è l insieme delle parole che contengono p solo alla fine.

Si considera una parola in S e si prova a concatenare la parola con una lettera del alfabeto. $S \times A$. La parola potrebbe non contenere ancora il pattern e quindi appartenere sempre a S oppure si potrebbe trovare il pattern alla fine cioè appartiene a T
$S \times A \cup \{\epsilon\} = S \cup T$
Concatenando invece della lettera si concatena tutto il pattern, ovviamente la parola avrà il pattern in fondo, ma potrebbe anche apparire prima nella parola. $S \times \{p\}$

$\overbrace{[\qquad\qquad\qquad}^{\text{S}}|p_1p_2 \ \dots\ p_{k-1}|p_{k+1} \dots p_k]$ Pattern in fondo
$\underbrace{[\quad p_1 \dots p_i|p_{i+1}p_{i+2} \dots p_k}_{\text{T}}\underbrace{|p_{k+1} \dots p_k]}_{\text{Coda dim=i}}$ Pattern nella parola

Si verifica quando $[p_1 \dots p_{k-1} = p_{i+1} \dots p_k] = c_i = 1$
$S \times \{p\} = T \times \cup_{c_i \neq 0} < coda >_i$
Si vogliono trasformare in funzioni generatrici.
Si chiamano $S(t)$ e $T(t)$
$S(t) + T(t) = S(t)mt + 1$
$S(t) \cdot t^k = T(t) \cdot c(t)$
L'unione della coda è direttamente $c(t)$

$T(t) = \frac{S(t)t^k}{c(t)}$
$S(t) + \frac{S(t)t^k}{c(t)} = S(t)mt + 1$

$C(t)S(t) + t^kS(t) = C(t)S(t)mt + c(t)$
$S(t) = \frac{C(t)}{t^k + c(t) - mtc(t)}$
$S(t) = \frac{C(t)}{t^k + (1 - mt)c(t)}$
$T(t) = \frac{t^k}{t^k + c(t)(1 - mt)}$

$L(t) = \frac{1}{1-mt} - S(t) = \frac{t^k + c(t)(1-mt) - (1+mt)c(t)}{(1-mt)(t^k + c(t)(1-mt))} = \frac{t^k}{(t^k + c(t)(1-mt))(1-mt)}$ funzione generatrice delle parole che contengono p

In cui $A^* = \frac{1}{1-a(t)} = \frac{1}{1-mt}$

$m = 2 \qquad S(t, w)$ funzione generatrice cdelle parole che evitano p in cui t tiene conto dei bit 1 e w tiene conto dei bit 0.
$S(t, w) + T(t, w) = S(t, w)(t + w) + 1$
$S(t, w)t^{n_i p} w^{n_o p} = T(t, w)c(t, w)$

$$\begin{array}{lcc} \text{aabbaa} & CodaPattern & esito \\ \text{aabba} & & 1\ C_0 \\ \text{aabba} & a & 0\ C_1 \\ \text{aabb} & aa & 0\ C_2 \\ \text{aab} & baa & 0\ C_3 \\ \text{aa} & bbaa & 1\ C_4 \\ \text{a} & abbaa & 1\ C_5 \end{array}$$

Invece di contare la lunghezza delle code, prendendo $a = 0, b = 1$,
$C(t, w) = 1 + t^2w^2 + t^2w^3$

$A = \{a, b, c\}$
$S ::= BC$
$B ::= aBb|c$
$C ::= aC|a$

Simbolo non terminale $\rightarrow$ f.g.
$::= \quad \rightarrow \quad =$
Simbolo terminale $\rightarrow$ la trasformazione dipende da cosa voglio contare
$| \rightarrow +$
concatenazione -> *

$S(t) = B(t)C(t) \Rightarrow S(t) = \frac{t^2}{(1-t)(1-t^2)}$
$B(t) = t^2B(t) + t \Rightarrow B(t) = \frac{t}{1-t^2}$
$C(t) = tC(t) + t \Rightarrow C(t) = \frac{t}{1-t}$

Se prendiamo t lunghezza, w numero di a

$$S(t,w) = B(t,w)C(t,w)$$
$$B(t,w) = t^2 wB(t,w) + t$$
$$C(t,w) = twC(t) + tw$$