# What are...zeta functions of languages?

Or: Counting voodoo

# The most classical zeta function

Ueber die Anzahl der Primzahlen unter einer gegebenen Grösse.

(Monatsberichte der Berliner Akademie, November 1859.)

Bei dieser Untersuchung diente mir als Ausgangspunkt die von Euler gemachte Bemerkung, dass das Product

$$\Pi \frac{1}{1-\frac{1}{p^s}} = \Sigma \frac{1}{n^s},$$

- The Riemann zeta function plays a pivotal role in mathematics

$$\zeta(s) = \frac{1}{1^s} + \frac{2}{2^s} + \frac{1}{3^s} + \ldots = \exp\left(\sum_{n=2}^{\infty} \frac{\Lambda(n)}{\log(n)} n^{-s}\right) = \prod_{prime} \frac{1}{1-p^{-s}}$$

- Today Focus on the discrete properties of zeta-type-functions

# Formal languages

| | *symbols* | *symbol name* | *string name* |
|---|---|---|---|
| *binary* | 01 (or ab) | bit | bitstring |
| *Roman* | abcdefghijklmnopqrstuvwxyz<br>ABCDEFGHIJKLMNOPQRSTUVWXYZ | letter | word |
| *decimal* | 0123456789 | digit | integer |
| *special* | ~`!@#$%^&*()_-+={[}]\|\:;”’<,>.?/ | | |
| *keyboard* | *Roman* + *decimal* + *special* | keystroke | typescript |
| *genetic code* | ATCG | nucleotide base | DNA |
| *protein code* | ACDEFGHIKLMNPQRSTVWY | amino acid | protein |

- $A^*$ = free monoid on the alphabet set $A$
- $L \subset A^*$ is a (formal) language
- The elements of $L$ are words

## A zeta function

$$\zeta_L(s) = \exp\left(\sum_{n=1}^{\infty} w_n \frac{s^n}{n}\right)$$

$w_n$ = number of words of length $n$

Example

- $A = \{0, 1\}$, $L = A^*$ is the language of binary strings
- Words are *e.g.* $\emptyset$, 0, 1, 00, 01, 10, 11, *etc.*
- $w_n = 2^n$
- So we get

$$\zeta_L(s) = \exp\left(\sum_{n=1}^{\infty} \frac{2^n s^n}{n}\right)$$

- The Taylor expansion is

$$\zeta_L(s) = 1 + 2s + 4s^2 + 8s^3 + \dots$$

## Enter, the theorem

For any cyclic language $L$ we have the Euler expansion

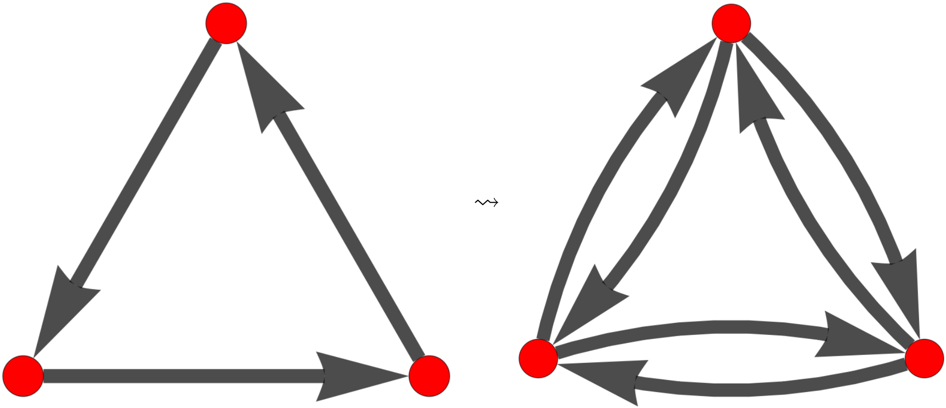
$$\zeta_L(s) = \prod_{n=1}^{\infty} \frac{1}{(1-s^n)^{c_n}}$$

$c_n = \#$(conjugacy classes of primitive words of length $n$ in $L$)

- Primitive = not a proper power, conjugate $u = rs$ and $v = sr$
- $L$ is called cyclic if $(uv \in L \Leftrightarrow vu \in L)$ and $(u \in L \Leftrightarrow u^n \in L$ for all $n)$
- $(uv \in L \Leftrightarrow vu \in L)$ "=" reading order does not matter
- $(u \in L \Leftrightarrow u^n \in L$ for all $n)$ "=" root closed
- Example The language of binary strings is cyclic and

$$\zeta_L(s) = \frac{1}{(1-s^1)^2} \cdot \frac{1}{(1-s^2)^1} \cdot \frac{1}{(1-s^3)^2} \cdot \frac{1}{(1-s^4)^3} \cdot \frac{1}{(1-s^5)^6} \cdot \ldots$$

## Counting on graphs



- For every graph $\Gamma$ add edges in reverse orientation and get $\Gamma^{\leftrightarrow}$
- $\Gamma^{\leftrightarrow}$ defines a cyclic language via paths
- The counting miracle $w_n$ "=" $c_n$ applies

**Thank you for your attention!**

---

I hope that was of some help.