# BeKo Modulklausur 12.01.2023

A NP-vollst $\Leftrightarrow$ alle Sprachen in NP auf A

<u>poly-Zeit</u> reduzierbar

+

$A \in NP$

Gedankenexperiment

Was wäre wenn $\leq_m^P$ ersetzen mit $\leq_m^{up}$

↓

Reduktion die vo-
nicht det. TM in Polynomzeit
berechnet werden kann

Welche Probleme sind NP-schwer unter „$\leq_m^{up}$"

$\forall_{L \in NP} \; L \leq_m^{up} SAT$

Satz v. Cook & Levin

Reduktionsfunktion $f$: bei Eingabe $x$
1. entscheide ob $x \in L$
2. Falls $x \in L$ Ausgabe $1 \rightarrow \in SAT$
sonst $0 \rightarrow \notin SAT$

$\forall_{L \in NP} \; L \leq_m^{up} X$

→ außer $\emptyset$ und $\Sigma^*$

→ alle Sprachen NP-schwer

# P-Vollständigkeit ?

A P-vollständig $\Leftrightarrow$ A P-schwer & $A \in P$

A P-schwer $\Leftrightarrow \forall_{L \in P} L \leq_u^L A$ → LOGSPACE

↝ HORNSAT

→ Gerhard Wöginger

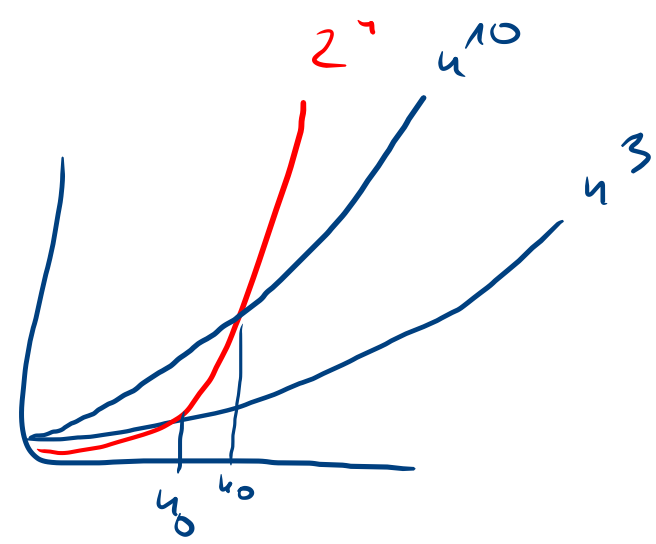
P vs NP „Beweise"

Algorithmus mit Laufzeit

$O(n \log^* n)$ oder $O(n \cdot \alpha(n))$

$\log^k n \leq c \cdot n \quad | 2^x$

$n^k \leq 2^c \cdot 2^n$




$(\log n)^{\text{acl}(n)}$

?

$\log(\text{acl}(n))$

$\downarrow$

schneller als

$\log(2^n) = n$

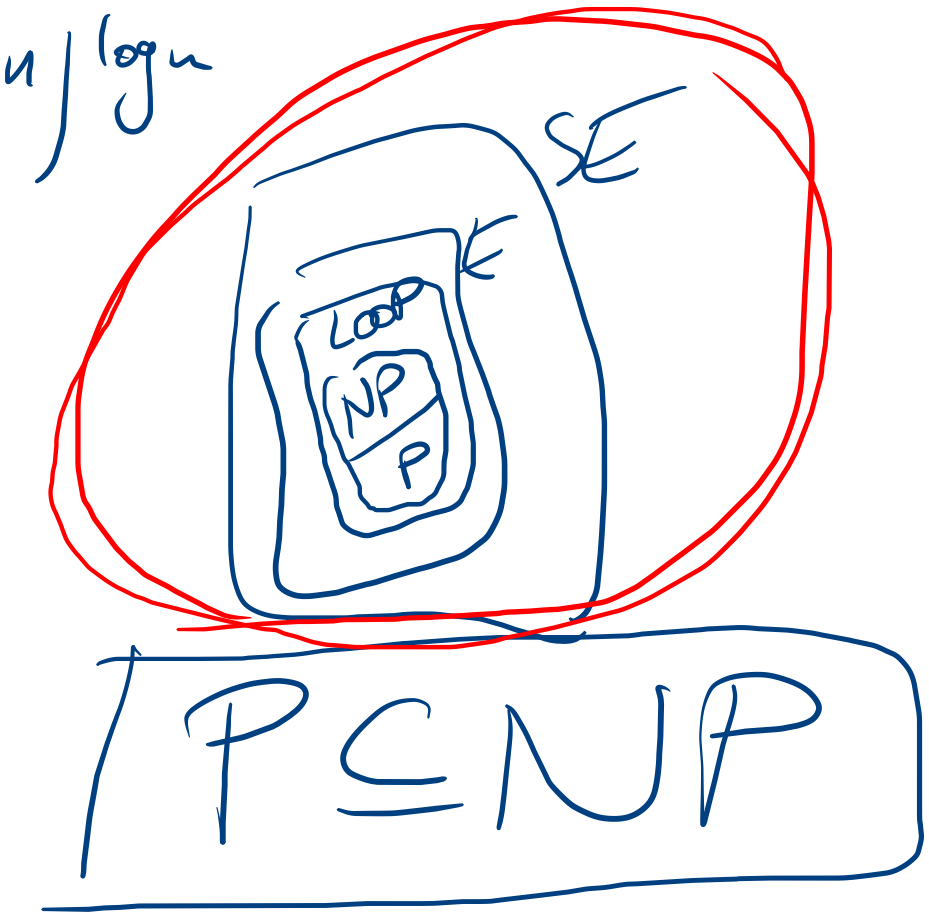
$(\text{acl}(n))^{\log n}$

$\downarrow$

wächst schneller als

$(2^n)^{\log n}$

→ wie wachsen die so im Vergleich?



#Zertifikate: $u \in \Sigma^{p(|x|)}$ $\quad |\Sigma| = c \in \mathbb{N}$

$$|\Sigma|^{p(|x|)} = c^{p(|x|)} = 2^{\log c \cdot p(|x|)} = 2^{q(|x|)}$$

$p, q$ Polynome

---

$2\text{-SAT} \leq_m^p SAT$

$f(\varphi) := \varphi$

$\varphi \in 2\text{-SAT} \Leftrightarrow \varphi$ erfüllbar $\Leftrightarrow \varphi \in SAT$

Wenn

$$3\text{-SAT} \leq_m^P 2\text{-SAT}$$

$$\in P$$

Dann 2-SAT NP-schwer !!!

Dann $P = NP$