$P \subsetneq EXPTIME$

Sei $L := \{ Kod(M) \# x \# 0^i \mid x \in \{0,1,\#\}^*, M$ hält auf x in höchstens 2^i Berechnungsschritten $\}$

$L \in EXPTIME$: Simuliere 2^i Schritte von M auf x ...

$L \notin P$: Angenommen, $L \in P$. Sei A ein Algo. mit $L(A) = L$ und $Time_A(n) \in O(n^c)$.
Konstruiere TM B:

- Prüft, ob Input die Form $Kod(M) \# 0^i$ hat, falls nein, verwirft
- Falls ja: simuliert A auf $Kod(M) \# Kod(M) \# 0^i \# 0^i$.
  - Falls A akz.: gehe in Endlosschlaufe
  - Falls A verw.: halte.

Betrachte Berechnung von B auf $Kod(B) \# 0^i$:

- Muss halten, denn falls nicht hält, muss A $Kod(B) \# Kod(B) \# 0^i \# 0^i$ akzeptieren, was äquiv. ist zu $Kod(B) \# Kod(B) \# 0^i \# 0^i \in L$, was wiederum äquiv. ist zu: "B hält auf $Kod(B) \# 0^i$ in 2^i Schritten."

- Also muss A $Kod(B) \# Kod(B) \# 0^i \# 0^i$ verwerfen
  - $\Rightarrow Kod(B) \# Kod(B) \# 0^i \# 0^i \notin L$
  - $\Rightarrow$ Berechnung von B auf $Kod(B) \# 0^i$ hat mind. 2^i Schritte.

Aber die Berechnung von B auf $Kod(B) \# 0^i$ hat nur $\underbrace{d_1(|Kod(B)| + 1 + i)}_{\text{Syntaxcheck}} + \underbrace{d_2(2 \cdot |Kod(B)| + 2i + 3)^c}_{\text{Simulation von } A}$ Schritte und

$$2^i > d_1(|Kod(B)| + 1 + i) + d_2(2 \cdot |Kod(B)| + 2i + 3)^c \quad \text{für } i \text{ gross genug} \quad ↯$$