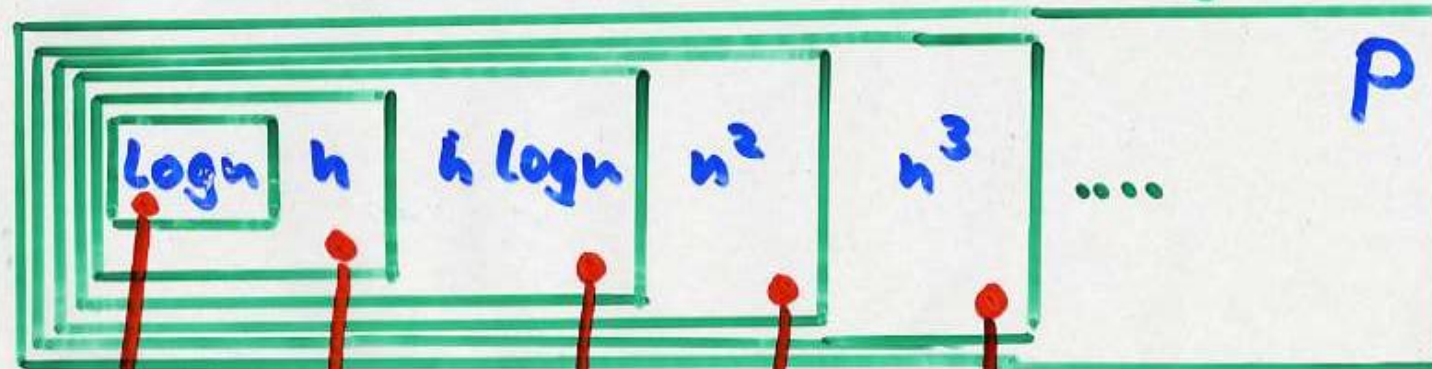


polynomieller Aufwand: die Klasse P

- Probleme in **P** können mit herkömmlicher Rechnertechnik in verfügbarer Zeit gelöst werden (wenn der Exponent nicht zu groß wird)



Suchen in balancierten Bäumen

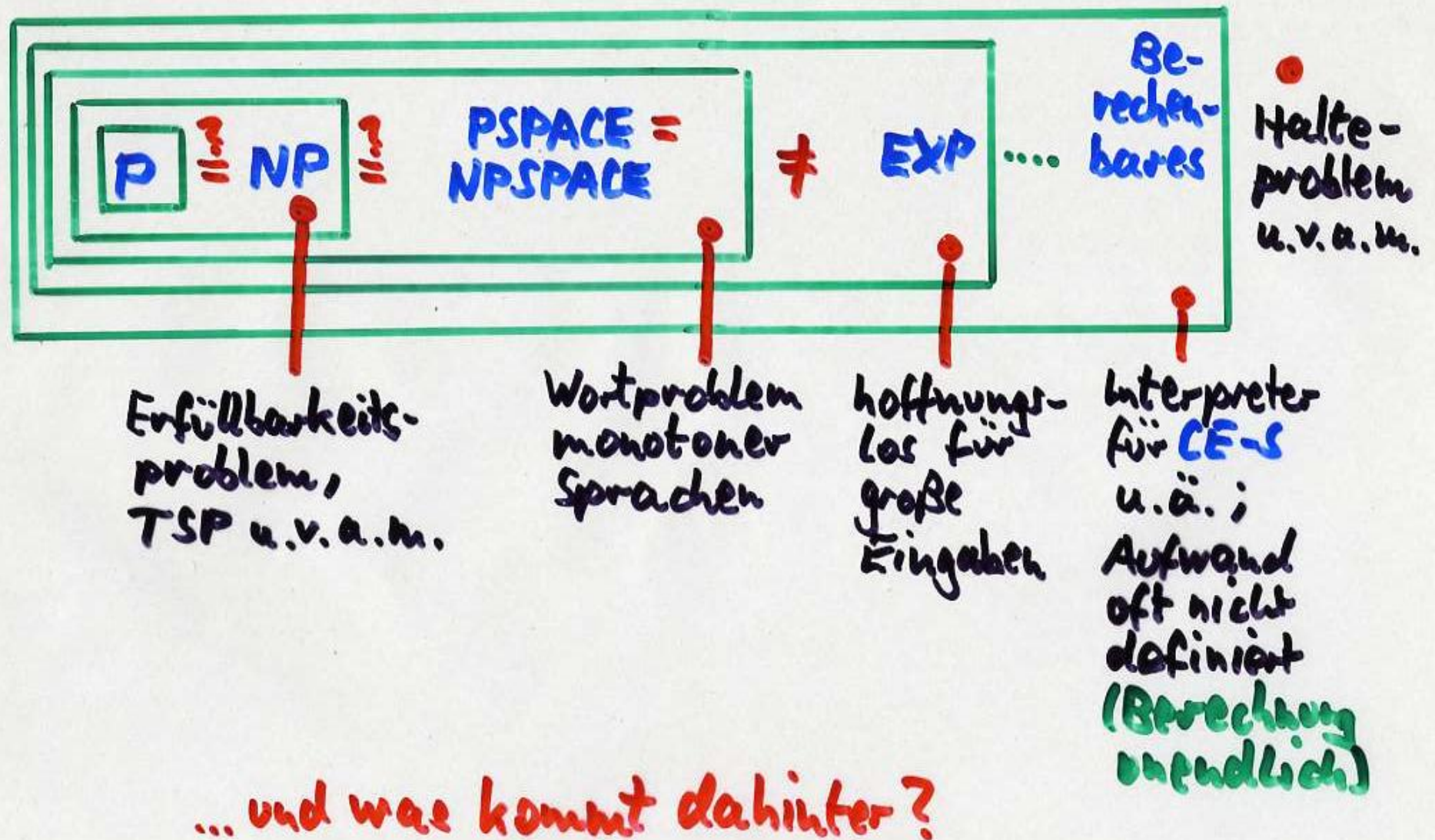
Suchen in Wörtern, Transponieren, Zählen, Filtern u.s.w.

Sortieren durch Mischen

Sortieren durch Einsortieren, quicksort

Matrizenmultiplikation; Wortproblem, kontextfreier Sprachen

wie weit reicht P?



polynomieller Platzbedarf

NPSPACE: Probleme mit nichtdeterministischen Lösungsalgorithmen, die polynomiellen Speicherplatz brauchen (im Verhältnis zur Größe der Eingabe)

PSPACE: analog für deterministische Lösungen

Beobachtung: $PSPACE \stackrel{(1)}{=} NPSPACE \stackrel{(2)}{\supseteq} NP \supseteq P$

(1) Beweis durch Backtracking

offenes Problem: $P \stackrel{?}{=} PSPACE$

typisches Beispiel: Wortproblem monotoner Sprachen

(2) konstanter Speicherplatzbedarf pro Rechenschritt

Chomsky-Grammatiken (Typ 0)

$G = (N, T, P, S)$

N : nichtterminale Zeichen
 T : terminale Zeichen
 P : Produktionen der Form $u ::= v$
 S : Startsymbol aus N

mit $N \cap T = \emptyset$

mit $u, v \in (N \cup T)^*$,
 $u \notin T^*$
(endlich viele)

- ▷ monoton: $\text{length}(u) \leq \text{length}(v)$ (Typ 1)
- ▷ kontextfrei: $\text{length}(u) = 1$ (Typ 2)
- ▷ regulär, rechtslinear: $\text{length}(u) = 1$ & $v \in T^* \cup T^+ N$ (Typ 3)

erzeugte Sprachen

▷ direkte Ableitung

$$xuy \xrightarrow{u \vdash v} xvy \quad \text{für alle } x, y \in (N \cup T)^*$$

(1) suche u als Teilwort eines Wortes

(2) ersetze u durch v

▷ Ableitung

$$w = w_0 \xrightarrow{r_1} w_1 \xrightarrow{r_2} w_2 \xrightarrow{r_3} \dots \xrightarrow{r_n} w_n = w'$$

dafür kurz auch: $w \xrightarrow[p]{n} w'$ oder $w \xrightarrow[p]{*} w'$

falls $r_1, \dots, r_n \in P$

▷ erzeugte Sprache

$$L(G) = \{w \in T^* \mid s \xrightarrow[p]{*} w\}$$

Wortprobleme

- ▷ Wortproblem von $L \subseteq T^*$: für alle $w \in T^*$ als Eingaben ist zu entscheiden, ob $w \in L$?
- ▷ Wortproblem $WP(G)$ einer Grammatik G ist Wortproblem von $L(G)$
- ▷ $WP(G) \in O(n^3)$ für kontextfreies G nach Cocke-Kasami-Younger
- ▷ $WP(G) \in O(n)$, falls deterministischer Kellerautomat K existiert mit $L(G) = L(K)$
- ▷ insbes. $WP(G) \in O(n)$ für rechtslineares G wegen endlicher Automaten

Erzeugung quadratisch langer Wörter

$G_{\text{square}} = (\{S, A, B, C, D, E, F\}, \{a\}, P_{\text{square}}, S)$ mit den Produktionen

$S ::= ABCD, BC ::= aCB, BD ::= ECCD, Aa ::= aA, Ca ::= aC, CE ::= EC,$
 $AE ::= AB, ABC ::= F, FC ::= F, FD ::= \lambda$

$S \rightarrow ABCD \rightarrow AaCBD \rightarrow aACBD \rightarrow aACECCD \rightarrow aAEC^3D \rightarrow aABC^3D \rightarrow$
 $\downarrow \text{FD}$
 $\downarrow$
 λ

$\downarrow$
 aFC^2D
 $\downarrow$
 aFD
 $\downarrow$
 a

$\rightarrow aAaCBC^2D \xrightarrow{2} aA(aC)^3BD \xrightarrow{5} a^4AC^3BD \rightarrow a^4AC^3EC^2D \xrightarrow{3} a^4AEC^5D \rightarrow a^4ABC^5D \rightarrow$
 $\downarrow$
 a^4FC^4D
 $\downarrow$
 a^4FD
 $\downarrow$
 a^4

$\rightarrow \dots \rightarrow a^{n^2} ABC^{2n+1} D \xrightarrow{*} a^{(n+1)^2} ABC^{2(n+1)+1} D \xrightarrow{*}$
 $\downarrow$
 a^{n^2}

$\downarrow$
 $a^{(n+1)^2}$

Wortproblem monotonen Grammatiken in PSPACE

Beobachtung: $WP(G) \in \text{PSPACE} = \text{PSPACE}$ für monotonen G

Beweis: Sei $G = (N, T, P, S)$ monotone Grammatik und $w_0 \in T^*$.

Betrachte folgenden nichtdeterministischen Algorithmus:

Konfigurationen: (w, w') mit $\text{Length}(w') \leq \text{Length}(w)$

Berechnungsschritt: $(w, w') \vdash (w, w'')$ falls $w' \xrightarrow{p} w''$

Start: (w_0, S)

Ziel: (w_0, w_0)

nichtdeterministisch

Platzbedarf: $2 \cdot \text{Length}(w_0)$

Korrektheit: $w_0 \in L(G)$ gdw. $(w_0, S) \vdash^* (w_0, w_0)$

Lemma: $w_1 \xrightarrow{p}^* w_2$ impliziert $\text{Length}(w_1) \leq \text{Length}(w_2)$

**dieses ist
aus meiner
Sicht das
beste Buch
über
theoretische
Informatik**

