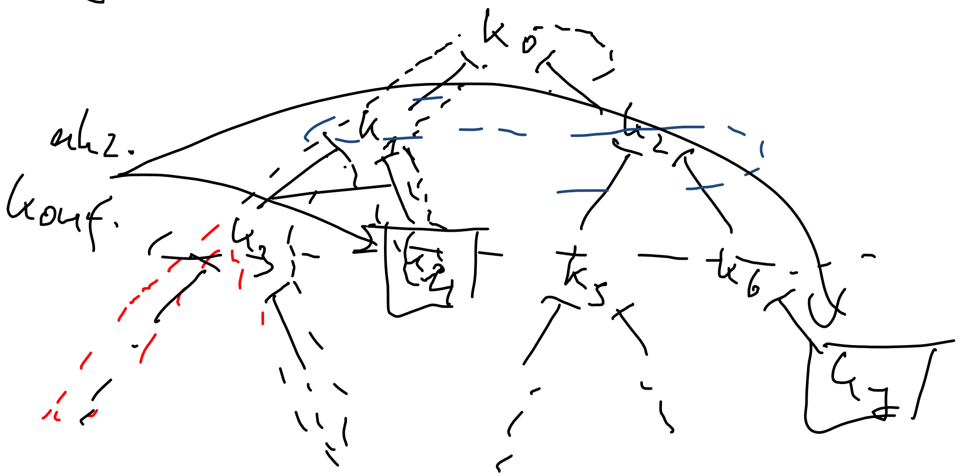
Komplexitätstheorie

Komplexitätsklasse

P $\quad$ $time_M(x)$ $\quad$ $TIME(f(x))$

$$P = \bigcup_{p \text{ Polynom}} TIME(p(n))$$

Erweiterung für nicht-det. TMs



Def Für eine nicht-det. TM M sei

$$\mathrm{ntime}_M(x) = \begin{cases} \min[\text{Länge einer akz. Rechnung}] & x \in T(M) \\ 0 & x \notin T(M) \end{cases}$$

Def Sei $f: \mathbb{N} \to \mathbb{N}$ eine Fkt. Die Klasse $\underline{N}TIME(f(n))$ besteht aus den Sprachen A, für die es eine nicht-det Mehrband-TM M gibt mit $A = T(M)$ und $\underline{n}\mathrm{time}_M(x) \le f(|x|)$.

Def $NP = \bigcup_{p \text{ Polynom}} NTIME(p(n))$

Folgerung: $P \subseteq NP$ $\qquad NP \overset{?}{\subseteq} P$

Simulation einer nicht-det. TM auf einer det. TM ist möglich, erfordert aber exponentiellen Zeitaufwand.

Beispiele:

Shortest Path Problem (SPP)

Gegeben: Graph $G = (V, E)$, $s, t \in V$, $k \in \mathbb{N}$.

Gefragt: Gibt es einen Pfad der Länge höchstens k von s nach t?

-> kann in $O(n^2)$ gelöst werden

-> $SPP \in P$

Longest Path Problem (LPP)

Gegeben: – wie oben –

Gefragt: Gibt es einen einfachen Pfad der Länge k oder länger von s nach t?

einfacher Pfad = kein Knoten darf zweimal vorkommen

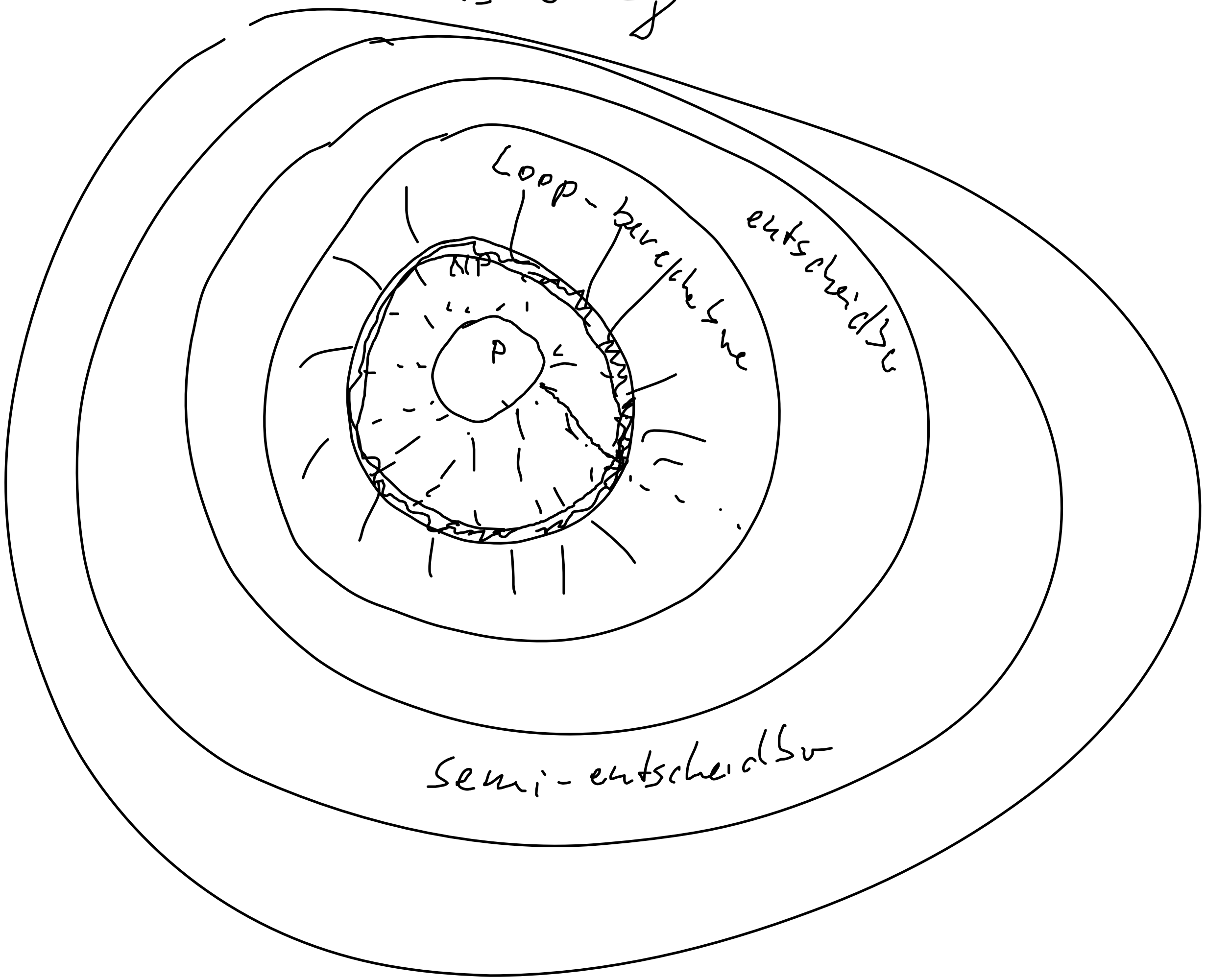
Nicht-det. Alg. für LPP:

s sei akt. Knoten
i := 0

WHILE t ist nicht akt. Knoten DO
  markiere akt. Knoten als "fertig"
  wähle nicht-deterministisch einen Nachbarknoten v
  IF v ist als fertig markiert THEN **FAIL**
  mache v zum aktuellen Knoten
  i := i+1
END
IF i ≥ k THEN **SUCCEED**
FAIL

Einfacher:
1. Wähle Sequenz von Knoten $v_1, \ldots, v_n$
2. Verifiziere, dass dies eine Lösung des LPP ist

Visualisierung



4.2 NP-Vollständigkeit

Def Seien $A \subseteq \Sigma^*$ und $B \subseteq \Gamma^*$ Sprachen. Dann heißt A auf B polynomial reduzierbar, symbolisch $A \leq_p B$, falls es eine totale und in polynomialer Laufzeit berechenbare Fkt $f: \Sigma^* \to \Gamma^*$ gibt, so dass für alle $x \in \Sigma^*$ gilt:

$$x \in A \text{ gdw } f(x) \in B$$

Lemma Falls $A \leq_p B$ und $B \in P$ ($B \in NP$), so ist auch $A \in P$ ($A \in NP$).

Bew: Sei $A \leq_p B$ mittels f gegeben, wobei M_f f in Zeit $p(n)$ berechnet. χ_B wird durch M_B in $q(n)$ Zeit berechnet.
$M_f ; M_B$ berechnet χ_A in Zeit

$$p(|x|) + q(\boxed{|f(x)|}) \leq \underbrace{p(|x|) + q(\boxed{p(|x|)})}_{\text{Polynom!}}$$

(p, q Polynome!)

für det. und nicht-det. Maschine. D.h. $A \in P$ ($A \in NP$). □

Bem: $\leq_p$ ist offensichtlich transitiv.

Frage: Wie findet man die schwersten Probleme in NP?

Def:

Eine Sprache A heißt NP-hart, falls für alle Spr. $L \in NP$ gilt $L \leq_p A$. Eine Spr. heißt NP-vollständig, falls A NP-hart ist und $A \in NP$.

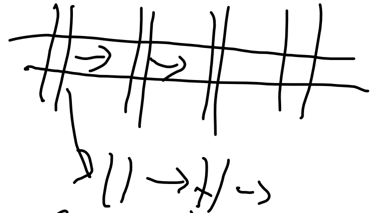
Satz: Sei A NP-vollständig. Dann gilt

$$A \in P \text{ gdw. } P = NP.$$

Bew: ($\Leftarrow$): Falls $P = NP$ folgt, dann folgt $A \in P$, da $A \in NP$.

($\Rightarrow$): Sei $A \in P$ und sei $L \in NP$ beliebig. Da A NP-hart, gilt $L \leq_p A$. Mit obigem Lemma folgt $L \in P$. Da L beliebig, gilt $P = NP$.

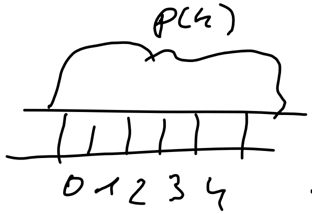
Frage: Gibt es solche NP-vollst. Sprachen?



Def: Das Erfüllbarkeitsproblem der Aussagenlogik (SAT) ist definiert durch

Gegeben: Eine Formel F der Aussagenlogik

Gefragt: Ist F erfüllbar? $x_0^0\ x_0^1\ x_0^2$



Satz (Cook)

SAT ist NP-vollständig.

$x_0\ x_1\ x_2$

$E_0\ E_1\ E_2$

$z_0\ z_1\ z_2\ z_3$

$k_0 = F$
$k_1 = T$
$k_2 = F$
$k_3 = F$

Bew:

SAT $\in$ NP: Rate Belegung. Überprüfe, ob Belegung die Formel wahr macht.

SAT ist NP-hart: Generische Reduktion!

Sei $L \in NP$ beliebig. Zu zeigen $L \leq_p$ SAT.

Sei M_L die nicht-det. TM, die L akzeptiert.

Sei $w \in \Sigma^*$ ⇝ Formel F_w konstr.

$\boxed{w \in L \text{ gdw } F_w \in SAT}$