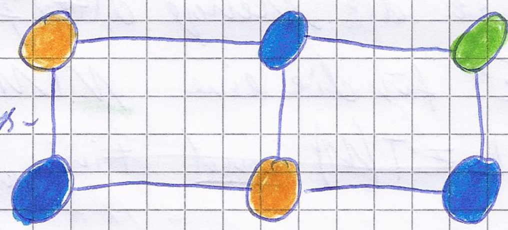
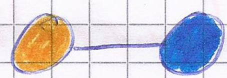
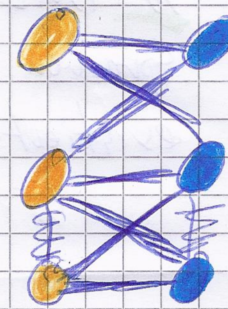
Theo12VL: III Komplexitätstheorie

28.06.11

Bsp. Graphfärbungsprobleme







3-Färbung $\Rightarrow$ NP-Schwer

2-Färbung

$\rightarrow$ „quantitative Berechenbarkeit"

III 1. Klassen

Def. Sei $f: \mathbb{N} \to \mathbb{N}$ monoton.

DTIME$(f(n))$ ist die Menge aller formalen Sprachen $L \subseteq \Sigma^*$ die von einer (Mehrband-) TM erkannt werden derart, dass für jedes $x \in \Sigma^*$ die TM maximal $f(\underbrace{|x|}_{=:n})$ Schritte ausführt.

Dazu: $time_M(x) :=$ Anzahl der Schritte von M auf x.

TM

Die Klasse P ist def. vermöge

$$\boxed{P := \bigcup_{k \geq 1} DTIME(n^k)}$$

Polynomielle Laufzeit!

Bsp. Sortieren: n^2

Achtung: Benutzen der O-Notation: konst. Faktoren werden vernachlässigt.

Def.:

$NTIME(f(n))$ sei die Menge aller formalen Sprachen $L \subseteq \Sigma^*$ für die eine NTM M existiert mit $L = T(M)$ und $time_M(x) \leq f(|x|)$ für alle x

→ Worst-Case-Komplexität

Wir setzen:

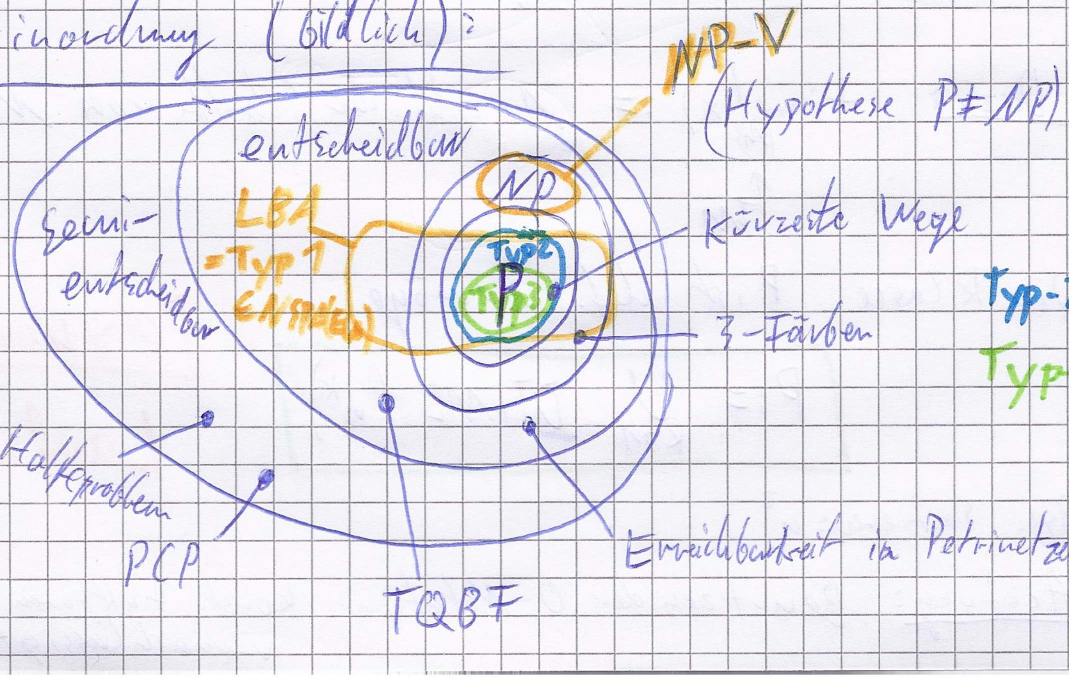
$$NP := \bigcup_{c \geq 1} NTIME(n^c)$$

Nichtdeterministisch Polynomiell

$$\boxed{\text{Zentral offen: } P \stackrel{?}{=} NP \quad (\text{per Def.: } P \subseteq NP)}$$

z.B. „Hat 3-Färbung von Graphen einen det. Polynomzeitalgorithmus?"

Einordnung (bildlich):



Typ-2: $CF \in DTIME(n^3)$

Typ-3: $REG \in DTIME(n)$

III 2 NP-Vollständigkeit

Def. Eine Sprache $A \subseteq \Sigma^*$ heißt polynomiell reduzierbar auf eine Sprache $B \subseteq \Pi^*$ (in Zeichen $A \leq_m^p B$) genau dann, wenn es eine totale, in polynomieller Zeit berechenbare Funktion

$f: \Sigma^* \to \Pi^*$ gibt, die

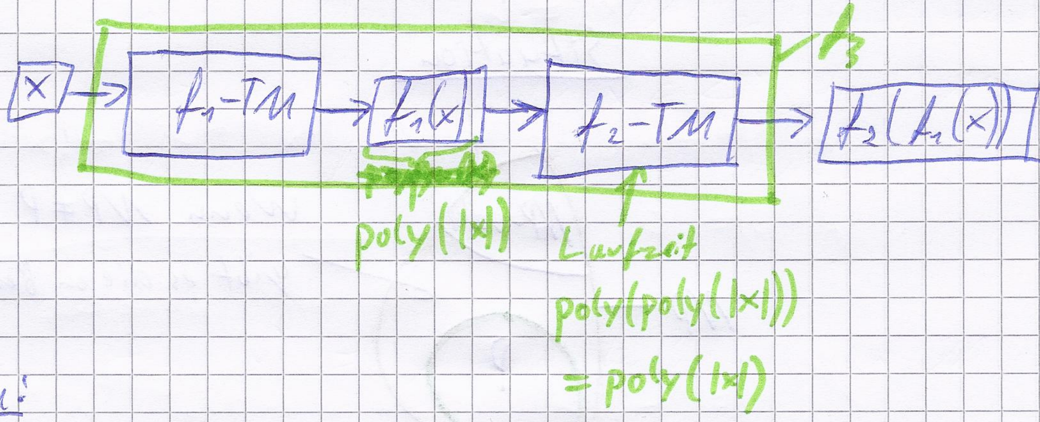
$$\boxed{\forall x \in \Sigma^*: x \in A \Leftrightarrow f(x) \in B.}$$

Hinweise:

a) $A \leq^p B \Rightarrow A \leq B$

b) $\leq^p$ ist transitiv

$(A \underset{f_1}{\leq^p} B, \; B \underset{f_2}{\leq^p} C \Rightarrow A \underset{f_3}{\leq^p} C$



Lemma: Wenn $A \leq^p B$ und $B \in P$ (bzw. $B \in NP$) so gilt $A \in P$ (bzw. $A \in NP$).

Beweis analog wie oben: (polynom von polynom ist ein Polynom!)

Def.: a) Ein Problem A heißt <u>NP-schwer</u> „NP-hard"
falls $\forall L \in NP$ gilt, dass $L \leq^P A$.

b) A heißt weiter $\boxed{\text{NP-vollständig}}$, wenn
A NP-schwer ist und $A \in NP$.

Hinweis: Ist A NP-schwer und $A \leq^P B$,
so ist auch B NP-schwer.

Satz: Ist A NP-vollständig, so gilt:

$$A \in P \iff P = NP$$

Beweis:

„$\Rightarrow$": $\forall B \in NP: B \leq^P A \overset{A \in P}{\Rightarrow} B \in P \Rightarrow NP \subseteq P$

„$\Leftarrow$": $A \in NP, P = NP \Rightarrow A \in P$ ∎

<u>Situation</u>