Beispiel 219 (Probleme in $\mathcal{P}$)

- 2-COLORING: Gegeben ein Graph $G = (V, E)$, ist $\chi(G) \leq 2$?

  2-COLORING $\in \mathcal{P}$, da $\chi(G) \leq 2$ gdw G bipartit.
- Shortest-Distance: Gegeben ein gewichteter Digraph $G = (V, E, d)$, $s, t \in V$ und eine Zahl D, ist die Entfernung von s nach t in $G \leq D$?

  Die Frage kann z.B. mit Dijkstra's Algorithmus in polynomieller Zeit entschieden werden.
- Maximaler Fluss: Gegeben ein gewichteter Digraph $G = (V, E, c)$ und $s, t \in V$ (c wird als Kapazität interpretiert), was ist der maximale Fluss von s nach t?

  Dafür gibt es zahlreiche (auf dem max-flow-min-cut Prinzip basierende) polynomielle Algorithmen.
- ...

Definition 220

Sei N eine nichtdeterministische Turingmaschine. Dann

1. $$\mathsf{TIME}_N(x) := \begin{cases} \text{minimale Anzahl der} \\ \text{Schritte, die } N \text{ benötigt,} & \text{falls } x \in L(N) \\ \text{um } x \in \Sigma^* \text{ zu akzeptieren} \\ \\ 0 & \text{sonst} \end{cases}$$
2. $\mathsf{NTIME}(f) :=$ Menge aller Sprachen, für die es eine nichtdeterministische Mehrband-Turingmaschine N gibt mit $\mathsf{TIME}_N(x) \leq f(|x|)$ für alle $x \in \Sigma^*$
3. $$\mathcal{NP} = \bigcup_{p \text{ Polynom}} \mathsf{NTIME}(p)$$

Man beachte, da wir bei nichtdeterministischen Maschinen i.W. an akzeptierenden Berechnungen interessiert sind, die etwas ungewöhnliche Festlegung für $x \notin L(N)$.

Eine alternative Definition für $\mathcal{NP}$ ist

Definition 221

$\mathcal{NP}$ ist die Klasse der Probleme L, für die es ein Prädikat $P \in \mathcal{P}$ und ein Polynom p gibt, so dass

$$(\forall x \in \Sigma^*)\,[x \in L \Leftrightarrow (\exists y \in \Sigma^*)[|y| \leq p(|x|) \wedge P(x,y)]]$$

Beispiel 222 (Probleme in $\mathcal{NP}$)

- k-COLORING: Gegeben ein Graph $G = (V, E)$, ist $\chi(G) \leq k$?

  k-COLORING ist in $\mathcal{NP}$, da eine nichtdet. TM eine gültige k-Färbung raten und dann verifizieren kann (in der alternativen Definition wäre y die Beschreibung einer solchen k-Färbung!)
- COLORING: Gegeben ein Graph $G = (V, E)$ und $k \in \mathbb{N}$, ist $\chi(G) \leq k$?

  Begründung i.W. wie oben; hier ist jedoch k Teil der Probleminstanz, also der Eingabe!
- SAT: Gegeben eine boolesche Formel F, hat F eine erfüllende Belegung?

  Eine nichtdet. TM kann einfach eine erfüllende Belegung, falls eine solche existiert, raten und verifizieren.

Eine der wichtigsten offenen Fragen der Informatik (und nicht nur der Theoretischen Informatik!) ist

$$\mathcal{P} =^{?} \mathcal{NP}$$

Eine der wichtigsten offenen Fragen der Theoretischen Informatik ist

Wie könnte man $\mathcal{P} \neq \mathcal{NP}$ beweisen?

2. $\mathcal{NP}$-Vollständigkeit

Definition 223

Seien $A \subseteq \Sigma^*$ und $B \subseteq \Gamma^*$ Sprachen. Dann heißt A polynomiell reduzierbar auf B (polynomiell many-one-reduzierbar; i.Z. $A \preceq_p B$ oder $A \preceq_p^{\text{m-1}} B$ oder auch $A \leq_p B$), falls es eine polynomiell berechenbare totale Funktion $f : \Sigma^* \to \Gamma^*$ gibt, so dass gilt

$$(\forall x \in \Sigma^*)[x \in A \Leftrightarrow f(x) \in B]$$

Bemerkungen:

1. Die Relation $\preceq_p$ ist transitiv, d.h.
$$A \preceq_p B \wedge B \preceq_p C \Rightarrow A \preceq_p C$$
2. $$A \preceq_p B \wedge B \in \mathcal{P} \Rightarrow A \in \mathcal{P}$$
3. $$A \preceq_p B \wedge B \in \mathcal{NP} \Rightarrow A \in \mathcal{NP}$$

Definition 224

1. Eine Sprache A heißt $\mathcal{NP}$-schwer (oder $\mathcal{NP}$-hart, engl. $\mathcal{NP}$-hard), falls für alle Sprachen B aus $\mathcal{NP}$ gilt:

$$B \preceq_p A$$

2. Eine Sprache A heißt $\mathcal{NP}$-vollständig (engl. $\mathcal{NP}$-complete), falls $A \in \mathcal{NP}$ und A $\mathcal{NP}$-schwer ist.

Intuitiv sind die $\mathcal{NP}$-vollständigen Probleme die "schwersten" Probleme in $\mathcal{NP}$.

$\mathcal{NP}$-harte Probleme sind "mindestens so schwer" wie alle Probleme in $\mathcal{NP}$, müssen aber selbst nicht in $\mathcal{NP}$ sein, können also noch viel höhere Komplexität haben.

Satz 225

SAT *ist $\mathcal{NP}$-vollständig.*

Beweis:

Offensichtlich ist SAT $\in \mathcal{NP}$.

Es bleibt zu zeigen, dass SAT $\mathcal{NP}$-hart ist, d.h. $\forall L \in \mathcal{NP}$
$L \preceq_p SAT$.

Wir tun dies, indem wir die Berechnung einer nichtdet. TM so in einer Formel kodieren, dass diese genau dann erfüllbar ist, wenn die Maschine in polynomiell vielen Schritten akzeptiert.

Beweis (Forts.):

Für jedes $L \in \mathcal{NP}$ gibt es eine Maschine N und ein Polynom p, so dass $(\forall x \in L)\,[\mathsf{TIME}_N(x) \leq p(|x|)]$.

Für die Reduktion zeigen wir jetzt, wie man daraus eine Formel $F = F(x)$ von in $|x|$ polynomieller Größe konstruiert, mit

$$x = x_1 \cdots x_n \in L \quad \text{gdw} \quad F \text{ ist erfüllbar}$$

Die Formel F besteht aus mehreren Teilen:

$$F = R \wedge A \wedge U \wedge E$$

Beweis (Forts.):

Wir führen folgende Variablen ein:

- $B^s_{i,t}$ – die i-te Zelle des Bandes enthält zum Zeitpunkt t das Symbol $s \in \Sigma$, Σ das Bandalphabet von N;
- Z^q_t – die Turingmaschine befindet sich zum Zeitpunkt t im Zustand $q \in Q$, Q die Zustandsmenge von N;
- P^i_t – der Schreib-/Lesekopf der Turingmaschine befindet sich zum Zeitpunkt t an Position i.

Für $|x| = n$ sei $\hat{t} = p(n)$. Dann ist $0 \leq t \leq \hat{t}$, $-\hat{t} \leq i \leq \hat{t}$, $q \in Q$ und $s \in \Sigma$.

Insgesamt haben wir also nur $O(\hat{t}^2) = O(p(n)^2)$ viele Variablen.

Beweis (Forts.):

Die Teilformel R gibt die Randbedingungen, dass

1. die Turingmaschine zu jedem Zeitpunkt t auf genau einem Feld i steht
$$\bigwedge_{0\le t\le \hat{t}} \left(\left(\bigvee_{-\hat{t}\le i\le \hat{t}} P_t^i \right) \wedge \left(\bigwedge_{i\neq j} P_t^i \Rightarrow \neg P_t^j \right) \right) \quad ,$$

2. die Turingmaschine zu jedem Zeitpunkt t in genau einem Zustand q ist
$$\bigwedge_{0\le t\le \hat{t}} \left(\left(\bigvee_{q\in Q} Z_t^q \right) \wedge \left(\bigwedge_{q\neq q'} Z_t^q \Rightarrow \neg Z_t^{q'} \right) \right) \quad \text{und}$$

3. zu jedem Zeitpunkt t an jeder Bandposition i genau ein Symbol s steht
$$\bigwedge_{\substack{0\le t\le \hat{t} \\ -\hat{t}\le i\le \hat{t}}} \left(\left(\bigvee_{s\in\Sigma} B_{i,t}^s \right) \wedge \left(\bigwedge_{s\neq s'} B_{i,t}^s \Rightarrow \neg B_{i,t}^{s'} \right) \right)$$

Beweis (Forts.):

Die Teilformel A gibt die Anfangsbedingung, dass zum Zeitpunkt 0 die Turingmaschine im Startzustand q_0 an Position 1 des Bandes ist und auf dem Band die Eingabe $x_1 \cdots x_n$ steht:

$$Z_0^{q_0} \wedge P_0^1 \wedge \bigwedge_{1 \le i \le n} B_{i,0}^{x_i} \wedge \bigwedge_{\substack{-\hat{t} \le i < 1 \vee \\ n < i \le \hat{t}}} B_{i,0}^{\square},$$

wobei $\square$ das Leerzeichen ist.

Beweis (Forts.):

Die Teilformel U stellt sicher, dass die Übergänge zwischen zwei Zeitpunkten t und $t+1$ entsprechend der Übergangsrelation $\delta : Q \times \Sigma \to 2^{Q \times \Sigma \times \{-1,0,1\}}$ der Turingmaschine sind.

Die Teilformel U ergibt sich aus der Konjunktion von

$$\left(\neg P_t^i \wedge B_{i,t}^s \Rightarrow B_{i,t+1}^s\right)$$

und

$$P_t^i \wedge \bigvee_{(q',s',d)\in\delta(q,s)} \left(B_{i,t}^s \wedge Z_t^q \Rightarrow B_{i,t+1}^{s'} \wedge Z_{t+1}^{q'} \wedge P_{t+1}^{i+d}\right)$$

für alle $0 \leq t < \hat{t}$ und alle $-\hat{t} \leq i \leq \hat{t}$.

Beweis (Forts.):

Die Teilformel E modelliert schließlich die Bedingung, dass die Turingmaschine einen akzeptierenden Zustand erreicht.
O.B.d.A. nehmen wir an, dass N nach Erreichen eines Endzustandes (also $q \in Q_f$) in diesem verbleibt, d.h. für alle $q \in Q_f$, $s \in \Sigma$ gilt $\delta(q, s) = (q, s, 0)$. Dann ist E gegeben durch:

$$\bigvee_{\substack{q \in Q_f \\ 0 \le t \le \hat{t}}} Z_t^q$$

Daher ergibt sich:

$$x \in L \text{ gdw } F \text{ ist erfüllbar.}$$

Beachte: Für $n = |x|$ hat F die Größe $O(p(n)^3)$. □