

# Tutorium Woche 10

Dominik Bruhn, Tutorium Nr. 9 + 11

20.01.2010

# Agenda

- 1 Organisation
- 2 Komplexitätstheorie
  - Wiederholung
- 3 Aufgabe 1
- 4 Aufgabe 2
- 5 Aufgabe 3
- 6 Aufgabe 4
- 7 Ende



# Komplexitätsklassen (Wiederholung)

## Definition (**P**)

Problem  $P \in \mathcal{P}$ , genau wenn gilt:

# Komplexitätsklassen (Wiederholung)

## Definition (P)

Problem  $P \in \mathcal{P}$ , genau wenn gilt:

$P$  kann in **polynominaler** Zeit von einer **deterministischen** Turingmaschine **entscheiden** werden.

# Komplexitätsklassen (Wiederholung)

## Definition (P)

Problem  $P \in \mathcal{P}$ , genau wenn gilt:

$P$  kann in **polynomineller** Zeit von einer **deterministischen** Turingmaschine **entscheiden** werden.

## Definition (NP)

Problem  $P \in \mathcal{NP}$ , genau wenn gilt:

# Komplexitätsklassen (Wiederholung)

## Definition (P)

Problem  $P \in \mathcal{P}$ , genau wenn gilt:

$P$  kann in **polynominaler** Zeit von einer **deterministischen** Turingmaschine **entscheiden** werden.

## Definition (NP)

Problem  $P \in \mathcal{NP}$ , genau wenn gilt:

$P$  kann in **polynominaler** Zeit von einer **nicht deterministischen** Turingmaschine **entschieden** werden.

# Komplexitätstheorie

## Definition (NP-Schwere)

Ein Problem  $F$  heißt  $\mathcal{NP}$ -schwer oder  $\mathcal{NP}$ -hart falls gilt:

# Komplexitätstheorie

## Definition (NP-Schwere)

Ein Problem  $F$  heißt  $\mathcal{NP}$ -schwer oder  $\mathcal{NP}$ -hart falls gilt:  
Alle  $Q$  aus  $\mathcal{NP}$  sind polynomiell reduzierbar auf  $F$ .  
 $F$  muss selbst nicht  $\in \mathcal{NP}$  sein!

# Komplexitätstheorie

## Definition (NP-Schwere)

Ein Problem  $F$  heißt  $\mathcal{NP}$ -schwer oder  $\mathcal{NP}$ -hart falls gilt:  
Alle  $Q$  aus  $\mathcal{NP}$  sind polynomiell reduzierbar auf  $F$ .  
 $F$  muss selbst nicht  $\in \mathcal{NP}$  sein!

## Definition (NP-Vollständigkeit)

Ein Problem  $F$  heißt  $\mathcal{NP}$ -vollständig falls gilt:

# Komplexitätstheorie

## Definition (NP-Schwere)

Ein Problem  $F$  heißt  $\mathcal{NP}$ -schwer oder  $\mathcal{NP}$ -hart falls gilt:  
Alle  $Q$  aus  $\mathcal{NP}$  sind polynomiell reduzierbar auf  $F$ .  
 $F$  muss selbst nicht  $\in \mathcal{NP}$  sein!

## Definition (NP-Vollständigkeit)

Ein Problem  $F$  heißt  $\mathcal{NP}$ -vollständig falls gilt:

- $F$  ist in der Klasse  $\mathcal{NP}$ , d.h.  $F \in \mathcal{NP}$  und
- $F$  ist  $\mathcal{NP}$ -schwer

# Aufgabe 1

## HALF-CLIQUE

*Gegeben:* Ein ungerichteter Graph  $G = (V, E)$

*Gesucht:* Gibt es eine Teilmenge  $V' \subseteq V$  mit

$\forall v, w \in V', v \neq w : (v, w) \in E$  und  $|V'| \geq |V|/2$

# Aufgabe 1

## HALF-CLIQUE

*Gegeben:* Ein ungerichteter Graph  $G = (V, E)$

*Gesucht:* Gibt es eine Teilmenge  $V' \subseteq V$  mit

$\forall v, w \in V', v \neq w : (v, w) \in E$  und  $|V'| \geq |V|/2$

Beweise, dass HALF-CLIQUE  $\mathcal{NP}$ -vollständig ist!

*Bekannt:* CLIQUE ist  $\mathcal{NP}$ -vollständig

# Aufgabe 1

## HALF-CLIQUE

*Gegeben:* Ein ungerichteter Graph  $G = (V, E)$

*Gesucht:* Gibt es eine Teilmenge  $V' \subseteq V$  mit

$\forall v, w \in V', v \neq w : (v, w) \in E$  und  $|V'| \geq |V|/2$

Beweise, dass HALF-CLIQUE  $\mathcal{NP}$ -vollständig ist!

*Bekannt:* CLIQUE ist  $\mathcal{NP}$ -vollständig

## CLIQUE

*Gegeben:* Ein ungerichteter Graph  $G = (V, E)$  und  $k \in \mathbb{N}$

*Gesucht:* Gibt es eine Teilmenge  $V' \subseteq V$  mit

$\forall v, w \in V', v \neq w : (v, w) \in E$  und  $|V'| \geq k$

# Aufgabe 2

## "Beweis" für $\mathcal{P} \neq \mathcal{NP}$

Betrachten Sie folgenden Algorithmus für SAT:

- Durchlaufe für die gegebene Formel  $\phi$  alle möglichen Belegungen der Variablen mit den Wahrheitswerten
- Akzeptiere  $\phi$ , wenn eine der durchlaufenen Belegungen  $\phi$  erfüllt

Dieser Algorithmus hat eine mit der Anzahl der Variablen exponentiell wachsende Laufzeit. Daher hat das Problem SAT einen exponentiellen Aufwand und kann nicht in  $\mathcal{P}$  liegen. Weil aber SAT in  $\mathcal{NP}$  liegt, muß also  $\mathcal{P} \neq \mathcal{NP}$  gelten.

Finde den Fehler in diesem "Beweis".

# Aufgabe 3

Zeige, dass es unter der Voraussetzung  $\mathcal{P} = \mathcal{NP}$  möglich ist, für eine aussagenlogische Formel  $\phi$  in polynomieller Zeit eine erfüllende Belegung der Variablen zu finden, falls eine solche Belegung existiert!

# Aufgabe 4

## Traveling Salesman (TSP)

*Gegeben:* Ein Graph  $G = (V, V \times V)$ ,  $k \in \mathbb{N}$

*Gesucht:* Gibt es einen einfachen Kreis  $C = (v_1, v_2, \dots, v_n, v_1)$ , sodass  $n = |V|$  und  $\sum_{(u,v) \in C} d(u,v) \leq k$ , wobei  $d(u,v)$  die Entfernung zwischen den Knoten  $u$  und  $v$  ist.

# Aufgabe 4

## Traveling Salesman (TSP)

*Gegeben:* Ein Graph  $G = (V, V \times V)$ ,  $k \in \mathbb{N}$

*Gesucht:* Gibt es einen einfachen Kreis  $C = (v_1, v_2, \dots, v_n, v_1)$ , sodass  $n = |V|$  und  $\sum_{(u,v) \in C} d(u,v) \leq k$ , wobei  $d(u,v)$  die Entfernung zwischen den Knoten  $u$  und  $v$  ist.

Zeige, dass TSP  $\mathcal{NP}$ -vollständig ist.

Bekannt: HAMILTONIAN-CIRCUIT ist  $\mathcal{NP}$ -vollständig.

# Aufgabe 4

## Traveling Salesman (TSP)

*Gegeben:* Ein Graph  $G = (V, V \times V)$ ,  $k \in \mathbb{N}$

*Gesucht:* Gibt es einen einfachen Kreis  $C = (v_1, v_2, \dots, v_n, v_1)$ , sodass  $n = |V|$  und  $\sum_{(u,v) \in C} d(u,v) \leq k$ , wobei  $d(u,v)$  die Entfernung zwischen den Knoten  $u$  und  $v$  ist.

Zeige, dass TSP  $\mathcal{NP}$ -vollständig ist.

Bekannt: HAMILTONIAN-CIRCUIT ist  $\mathcal{NP}$ -vollständig.

## HAMILTONIAN-CIRCUIT

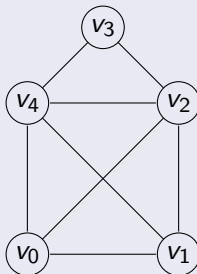
*Gegeben:* Ein ungerichteter Graph  $G = (V, E)$ .

*Gesucht:* Besitzt  $G$  einen Hamiltonkreis? (Dies ist eine Permutation  $\pi$  der Knotenindizes  $(v_{\pi(1)}, v_{\pi(2)}, \dots, v_{\pi(n)})$ , sodass für  $i = 1, \dots, n-1$  gilt:  $\{v_{\pi(i)}, v_{\pi(i+1)}\} \in E$ ) und außerdem  $\{v_{\pi(n)}, v_{\pi(1)}\} \in E$ ).

# Aufgabe 4

## Aufgabe 4.2

Gegeben sei folgender Graph:



Gibt es einen Hamiltonkreis? Wandel hierzu das Problem in ein TSP um und finde eine optimale Rundtour.

# Ende

# Fragen?

*Do not mistake coincidence for fate.*