Komplexität - Klassifikation von Problemen

Klassifikation von Problemen anhand der dazu bekannten Algorithmen

$$P \subseteq NP \subseteq EXPTime$$

- P: (höchstens) polynomialen Zeitaufwand
- EXPTime: (höchstens) exponentiellen Zeitaufwand
- NP: Nichtdeterministisch polynomialer Zeitaufwand
  - Lösung raten (exponentiell viele Alternativen)
  - Prüfen der Lösung mit polynomialem Aufwand

Was sind Probleme und welche sind in P??

- Zusammenhang Mengen und Prädikate
  - Entscheidungs-**Problem**: $x \in A$
  - Charakteristische Funktion zu A:

$$\chi_A(x) = \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{sonst} \end{cases}$$

  - A entscheidbar genau dann, wenn χ_A berechenbar
  - A in **P** genau dann, wenn χ_A in **polynomialer Zeit** berechenbar

- Beispiele
  - A_1= Menge der Primzahlen = $\{ x \in \mathbb{N} \mid x \text{ ist prim}\}$
  - A_2= Menge der Quadratzahlen = $\{ x \mid \text{es gibt } n \in \mathbb{N}: \ x=n^2\}$
  - A_3= Graph der Quadratfunktion = $\{ (x, y) \mid y= x^2 \}$
  - A_4= Graph sortierte Listen = $\{ (L, L') \mid L' = L \text{ sortiert} \}$

Komplexität NP: Formale Definition

- Es genügt Teilmengen von {0,1}* bzw. |N zu betrachten
  - Betrachte $x \in A$ für beliebige Menge A
  - x darstellbar auf Rechner als Binärstring „00010100 …10“
  - „00010100 …10“ darstellbar als Dualzahl 100010100 …10“

- $A \subseteq \{0,1\}^*$ ist in NP $\Leftrightarrow$
  - Es gibt ein in polynomialer Zeit p berechenbares Prüfprogramm P mit
  - $x \in A \;\Leftrightarrow\; \exists z \in \{0,1\}^*: \; |z| \leq p(|x|) \wedge P(x,z)$
  - Hinweis: |x| = Länge von x = Anzahl Bits von x

- Beispiel Erfüllbarkeitsproblem für Formeln der Aussagenlogik
  - z rät Belegung
  - P prüft ob die Formel x mit Belegung z erfüllt wird

Komplexität NPC: Formale Definition

- Menge B reduzierbar auf A (kurz: $B \leq_p A$) $\Leftrightarrow$
  - Es gibt in polynomialer Zeit berechenbare Funktion f mit
    $\forall x: \quad x \in B \Leftrightarrow f(x) \in A$
  - d.h. der Test „$x \in B$“ lässt sich berechnen durch „$f(x) \in A$“
    $\forall x: \quad \chi_B(x) := \chi_A(f(x))$

- $A \subseteq \{0,1\}^*$ ist NP-hart $\Leftrightarrow$
  - Jede Menge B in NP lässt sich auf A reduzieren
  - Kurz: $\forall B \in NP \quad B \leq_p A$
  - Lässt sich A in polynomialer Zeit berechnen, dann auch jedes B in NP

- $A \subseteq \{0,1\}^*$ ist NP-vollständig $\Leftrightarrow$
  - A ist in NP und
  - A ist NP-hart

Nachweis für NP-vollständig

- Nachweis für $A \subseteq \{0,1\}^*$ NP-vollständig
  - A ist in NP und
  - B ist NP-vollständig
  - $B \leq A$

- Beispiele für NP-vollständige Probleme B:
  - TSP
  - Bin-Packing
  - Knappsack

Komplexität – NPC-Probleme

- KP – Rucksackproblem.
  - n Gegenstände mit Gewicht g_1 .. g_n $\in$N und deren Wert w_1 .. w_n $\in$N , Maximale Traglast des Rucksacks g $\in$N, Kostengrenze K
  - Können Gegenstände unter Beachtung der Traglast in den Rucksack gepackt werden, dass deren Gesamtwert mindestens K ist?
  - Damit Optimierungsproblem berechenbar:

    Beladung des Rucksacks mit maximalem Wert

  Verallgemeinerung Bin Packing: m Bins = Rücksäcke mit Traglastgrenze g.

- TSP – Problem des Handlungsreisenden
  - n Orte, Kostengrenze K,

    n x n Kostenmatrix C = (c_{ij}) Entfernung von i nach j.
  - Gibt es eine Rundreise durch alle Orte, die Grenze K nicht übersteigt?
  - Damit Optimierungsproblem berechenbar:

    Billigste Rundreise durch alle n Orte

NP-Vollständigkeitsbeweise

- SAT = Erfüllbarkeitsproblem NP-vollständig

- Beweiskette
  - SAT $\leq_p$ 3SAT (Klauseln mit 3 Variable)
  - 3SAT $\leq_p$ HC (Hamilton-Kreis)
  - HC $\leq_p$ TSP
  - TSP $\leq_p$ $TSP_{Anwendung}$

Hinweis: 3SAT $\leq_p$ HC

https://opendsa-server.cs.vt.edu/ODSA/Books/Everything/html/threeSAT_to_hamiltonianCycle.html

SAT $\leq_p$ 3SAT

2SAT in P !!

$\Sigma^*_{SAT} \to \Sigma^*_{3-SAT}$ mit Übersetzungsregeln

→ Einzelne Betrachtung jeder Klausel

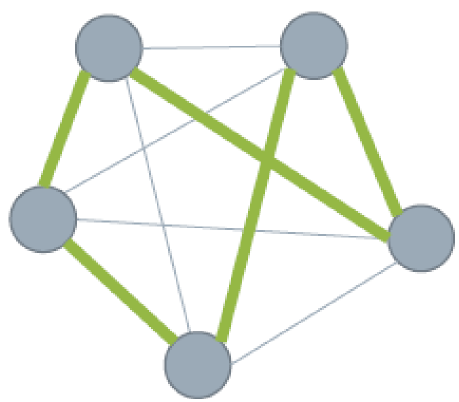
#Literale	SAT	3 – SAT
1	(z_1)	$(z_1 \vee z_1 \vee z_1)$
2	$(z_1 \vee z_2)$	$(z_1 \vee z_1 \vee z_2)$
3	$(z_1 \vee z_2 \vee z_3)$	$(z_1 \vee z_2 \vee z_3)$
> 3	$(z_1 \vee z_2 \vee z_3 \ldots \vee z_k)$	$(z_1 \vee z_2 \vee y_{c,1})$ $(\overline{y_{c,l}} \vee z_{l+2} \vee y_{c,l+1})$ für $1 \leq l \leq k-4$ $(\overline{y_{c,k-3}} \vee z_{k-1} \vee z_k)$
z.B. 7	$(z_1 \vee z_2 \vee z_3 \vee z_4 \vee z_5 \vee z_6 \vee z_7)$	$(z_1 \vee z_2 \vee y_{c,1})$ $(\overline{y_{c,1}} \vee z_3 \vee y_{c,2})$ $(\overline{y_{c,2}} \vee z_4 \vee y_{c,3})$ $(\overline{y_{c,3}} \vee z_5 \vee y_{c,4})$ $(\overline{y_{c,4}} \vee z_6 \vee z_7)$
	Genau dann erfüllbar, wenn mindestens ein $z = 1$	Genau dann erfüllbar, wenn mindestens ein $z = 1$

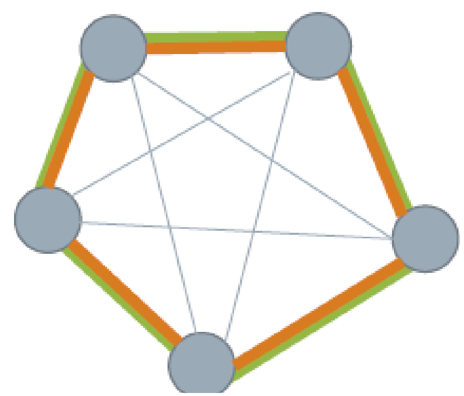
HC $\leq_p$ TSP

Hamiltonian Circuit $\leq_p$ Traveling Salesperson Problem

HC $\leq_p$ TSP

Enthält ein Graph <u>einen</u> Kreis, der jeden Knoten genau einmal berührt? $\leq_p$ Enthält ein Graph <u>einen durch c beschränkten</u> Kreis, der jeden Knoten genau einmal berührt?





Idee der Reduktion

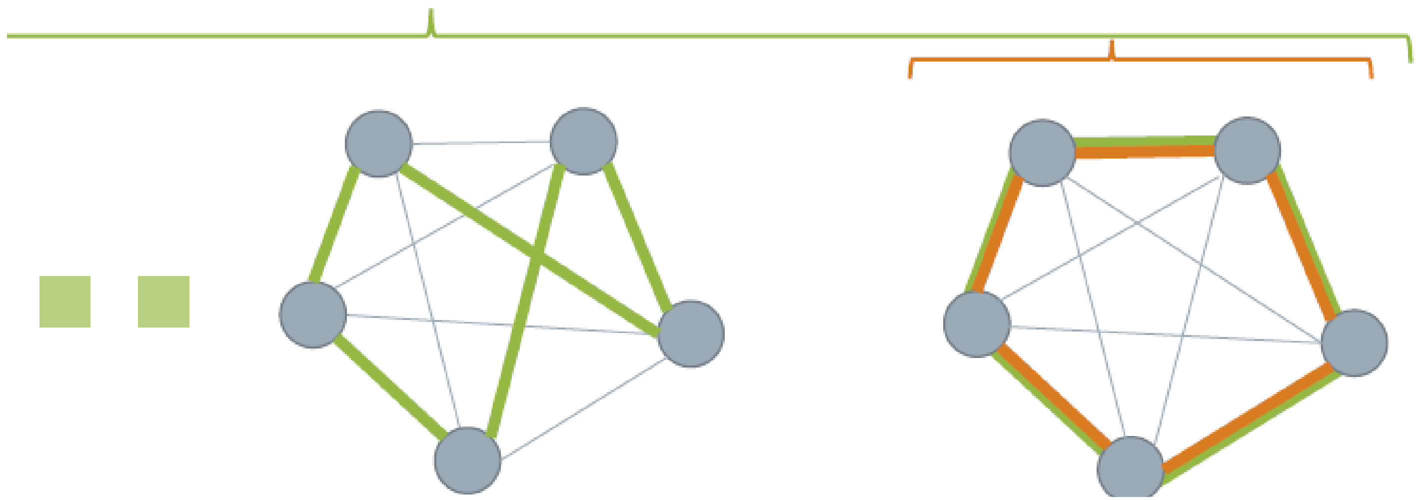
- HC-Kanten -> TSP Kanten mit Länge 1
- TSP Graph vervollständigen mit Kanten Länge 2
- HC mit n Knoten erfüllbar ⇔ TSP mit Route Länge n erfüllbar

HC $\leq_p$ TSP

Hamiltonian Circuit $\leq_p$ Traveling Salesperson Problem

HC $\leq_p$ TSP

Enthält ein Graph einen Kreis, der jeden Knoten genau einmal berührt? $\leq_p$ Enthält ein Graph einen durch c beschränkten Kreis, der jeden Knoten genau einmal berührt?



TSP nicht approximierbar

- HC-Kanten -> TSP Kanten mit Länge 1
- TSP Graph vervollständigen mit Kanten Länge **n^2**
- HC mit n Knoten erfüllbar ⇔ TSP mit Route Länge n erfüllbar