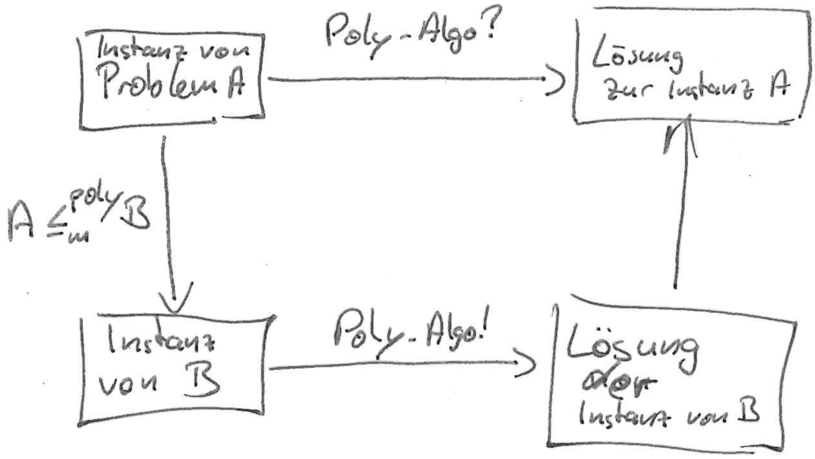
Bsp Abbildung Reduktionen



Ist $B \in P$, dann muss damit auch $A \in P$ sein.

Wüssten wir aber, dass A keinen Polyzeit-Algo erlaubt, dann kann damit auch B keinen Polyzeit-Algo erlauben!

Beweis Lemma 9.8

Zunächst: Es gibt nur polynomiell in n (Größe der Eingabe) viele Konfigurationen:

$$\underbrace{|Q|}_{\text{Kontrollzustände}} \cdot \underbrace{k|\Gamma|^{d'\log n + d''}}_{\text{Inhalt Arbeitsbänder}} \cdot \overbrace{n}^{\text{Kopfposition Eingabe}} \cdot \underbrace{kd'\log n + d''}_{\text{Kopfpositionen auf Arbeitsbändern}}$$

Da $|Q|, k, |\Gamma|, d', d''$ Konstanten (unabhängig von der Eingabe), lässt sich das Produkt zu $O(n^d)$ für ein $d \in \mathbb{R}_{>0}$ zusammenfassen.

$\Rightarrow$ Die Ausgabe kann höchsten $O(n^d)$ Zeichen umfassen.

Problem: Wir können $f(x)$ nicht auf dem Arbeitsband stehen haben.

$\Rightarrow$ Berechne die i-te stelle on-demand!

M_i sei eine TM, die die i-te Stelle von $f(x)$ berechnet.

Diese existiert auch:

1. Erstelle einen counter mit Wert i
2. Verhalte dich wie M_f um $f(x)$ zu berechnen.
   ↳ Statt Ausgaben, verringere den Counter
   ↳ Ist Counter $= 0$, speichere die Ausgabe auf dem Arbeitsband.

$\Rightarrow$ Damit existiert auch eine Maschine $M_{g\circ f}$, die sich wie M_g verhält um $g(f(x))$ zu berechnen. Benötigt M_g die i-te Stelle von $f(x)$, lassen wir aber zusätzlich M_i laufen.

Da $i \leq c \cdot n^d$ für $c, d \in \mathbb{R}_{>0}$ benötigen Counter für i nur $\log(c \cdot n^d) \in \mathcal{O}(\log n)$ Platz $\square$

(Für etwas mehr Details, siehe Skript)

Beispiele:
Betrachte zwei Probleme

1. USTCON:
   Gegeben: Ungerichteter Graph $G=(V,E)$, Knoten $s,t \in V$.
   Frage: Existiert ein Pfad von s nach t.

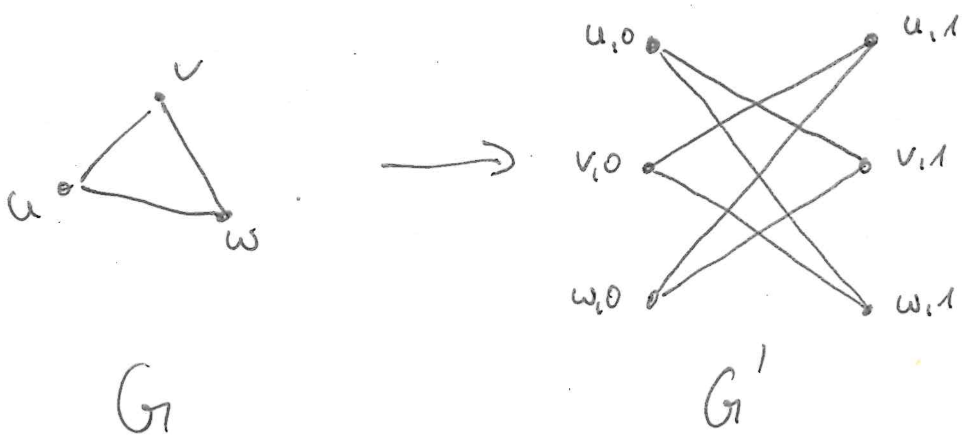
2. Odd-Cycle:
   Gegeben: Ungerichteter Graph $G=(V,E)$,
   Frage: Besitzt G einen ungeraden Kreis?

Man kann zeigen: USTCON $\in L$

Frage: Ist Odd-Cycle $\in L$?

Ja, wenn Odd-Cycle $\leq_m^{\log}$ USTCON.

Betrachte folgende Reduktion:
Sei $G'=(V',E')$ ein ungerichteter Graph mit $V'=V\times\{0,1\}$. Für jede Kante $\{u,v\}\in E$ füge Kanten $\{(u,0),(v,1)\}$, sowie $\{(v,0),(u,1)\}$ hinzu.



Claim:
Sei $s \in V$. G besitzt ungeraden Kreis C mit $s \in C$, gdw. G' einen $(s,0)$-$(s,1)$-Pfad besitzt.

"$\Rightarrow$" Ex. ungerader Kreis $(s, v_0, v_1, \ldots, v_k)$ mit $k \bmod 2 = 1$, in G, dann existiert in G' die Knotenfolge

$$(s,0), (v_0,1), (v_1,0), \ldots, (v_k,0), (s,1)$$

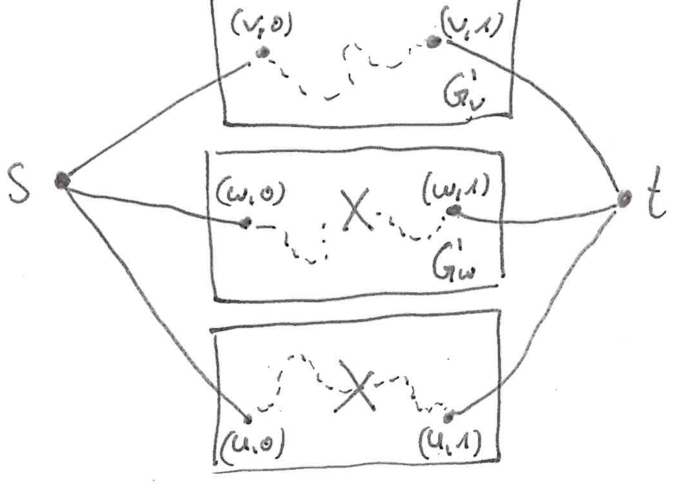
$\Rightarrow \exists\ (s,0)$-$(s,1)$-Pfad.

"$\Leftarrow$": Ang. es ex. ein solcher Pfad. Sei

$P = ((s,0), (v_0,1), (v_1,0), \ldots, (v_k,0), (s,1))$. ~~so kurz wie möglich~~

Dann besitzt P gerade viele Knoten (auf jede 0 folgt eine 1). Da $(s,0) \equiv (s,1)$ in G, muss der erhaltende Kreis $(s, v_0, v_1, \ldots, v_k)$ ungerade Länge besitzen.

Nun muss s nicht auf dem Kreis liegen!
Sei G'_v der konstruierte Graph, wie oben beschrieben.
Erstelle G'_v für jeden $v \in V$, füge Knoten s und t hinzu, sowie Kanten $\{s, (v,0)\}$ und $\{t, (v,1)\}$ mit $(v,0), (v,1) \in G'_v$



Selbst zu zeigen:

1. G enthält ungeraden Kreis $\Leftrightarrow$ neuer Graph besitzt s-t-Pfad.
2. Reduktion läuft mit log-space!