$\mathcal{NP}$-Vollständigkeit

Definition $\mathcal{NP}$-vollständig

Sei L eine Sprache. Wir bezeichnen L als $\mathcal{NP}$-vollständig, falls

1. $L \in \mathcal{NP}$
2. Für **jede** Sprache $A \in \mathcal{NP}$ gilt: $A \leq_p L$.

Separation oder Gleichheit von $\mathcal{P}$ und $\mathcal{NP}$

Satz

Sei L eine $\mathcal{NP}$-vollständige Sprache und $L \in \mathcal{P}$. Dann gilt $\mathcal{P} = \mathcal{NP}$.

Beweis:

- Wir zeigen für ein beliebiges $A \in \mathcal{NP}$, dass $A \in \mathcal{P}$.
- Da $A \in \mathcal{NP}$ und L $\mathcal{NP}$-vollständig ist, gilt $A \leq_p L$.
- Nach Voraussetzung gilt $L \in \mathcal{P}$.
- $\mathcal{P}$-Reduktionssatz: Aus $A \leq_p L$, $L \in \mathcal{P}$ folgt $A \in \mathcal{P}$.
- Da dies für ein beliebiges $A \in \mathcal{NP}$ gilt, folgt $\mathcal{NP} \subseteq \mathcal{P}$.
- Wegen $\mathcal{P} \subseteq \mathcal{NP}$ gilt schließlich $\mathcal{P} = \mathcal{NP}$.

$\mathcal{NP}$ Vollständigkeits-Beweise

Satz $\mathcal{NP}$-Reduktionssatz

Seien B, L Sprachen. Sei L $\mathcal{NP}$-vollständig, $B \in \mathcal{NP}$ und $L \leq_p B$.
Dann ist auch B $\mathcal{NP}$-vollständig.

Beweis: Müssen zeigen, dass $A \leq_p B$ für alle $A \in \mathcal{NP}$.

- Da L $\mathcal{NP}$-vollständig ist, gilt $A \leq_p L$ für beliebiges $A \in \mathcal{NP}$.
- Ferner gilt nach Voraussetzung $L \leq_p B$.
- Aus der Transitivität von $\leq_p$ folgt: $A \leq_p B$.
- Damit ist B ebenfalls $\mathcal{NP}$-vollständig.

Problem: Wir benötigen ein *erstes* $\mathcal{NP}$-vollständiges Problem.

Satz von Cook-Levin (1971)

Satz von Cook-Levin

SAT ist $\mathcal{NP}$-vollständig.

Beweis: Müssen zeigen

1. SAT $\in \mathcal{NP}$ (bereits gezeigt)
2. Für alle $L \in \mathcal{NP}$ existiert polynomiell berechenbare Reduktion f:
$$w \in L \Leftrightarrow f(w) \in \text{SAT}.$$

Beweisidee: Sei $L \in \mathcal{NP}$ beliebig.

- $\exists$ NTM N mit polynomieller Laufzeit n^k mit
$$w \in L \Leftrightarrow N \text{ akzeptiert } w.$$
- Konstruieren aus (N, w) eine Formel ϕ mit
  1. N akzeptiert w $\Leftrightarrow$ $f(w) = \phi \in$ SAT
  2. f ist in Zeit polynomiell in $|w| = n$ berechenbar.
- Betrachten dazu $(n^k + 1) \times (n^k + 1)$ Berechnungstabelle von N.

Berechnungstabelle T von N auf w

q_0	$\triangleright$	w_1	$\ldots$	w_n	$\sqcup$	$\ldots$	$\sqcup$
$\triangleright$	q_i	w_1 $\vdots$	$\ldots$	w_n	$\sqcup$	$\ldots$ $\vdots$	$\sqcup$

- Tabelle T entspricht einem Pfad im Berechnungsbaum.
- Erste Zeile enthält die Startkonfiguration.
- $(i+1)$-te Zeile ist mögliche Nachfolgekonfiguration der i-ten Zeile.
- In Laufzeit n^k können höchstens n^k Zellen besucht werden.
- T akzeptierend $\Leftrightarrow$ T enthält eine akzeptierende Konfiguration.
- Konstruieren ϕ derart, dass ϕ erfüllbar ist gdw. T akzeptierend ist.

Struktur der Formel für ϕ

- Sei $T(i,j)$ der Eintrag in der i-ten Zeile und j-ten Spalte von T.
- $T(i,j) \in Q \cup \Gamma$ für alle i,j.
- Definieren ϕ über den Booleschen Variablen $x_{i,j,s}$ mit

$$x_{i,j,s} = 1 \Leftrightarrow T(i,j) = s \quad \text{für } s \in Q \cup \Gamma.$$

Formel für ϕ: $\phi = \phi_{Start} \wedge \phi_{accept} \wedge \phi_{Eintrag} \wedge \phi_{move}$ mit

ϕ_{Start}: T beginnt mit Startkonfiguration.

ϕ_{accept}: T muss Eintrag q_a besitzen.

$\phi_{Eintrag}$: T enthält Einträge aus $Q \cup \Gamma$.

ϕ_{move}: T besitzt gültige Nachfolgekonfigurationen.

Definition von ϕ_{Start}, ϕ_{accept} und $\phi_{Eintrag}$

ϕ_{Start}: Kodieren die Startkonfiguration $q_0 \rhd w_1 \dots w_n$

$$x_{1,1,q_0} \wedge x_{1,2,\rhd} \wedge x_{1,3,w_1} \wedge \dots \wedge x_{1,n+2,w_n} \wedge x_{1,n+3,\sqcup} \wedge \dots \wedge x_{1,n^k+1,\sqcup}$$

ϕ_{accept}: ϕ ist erfüllend gdw T eine erfüllende Konfiguration enthält

$$\phi_{accept} = \bigvee_{1 \leq i,j \leq n^k+1} x_{i,j,q_a}$$

$\phi_{Eintrag}$: $T(i,j) \in Q \cup \Gamma$, d.h. es gibt ein $s \in Q \cup \Gamma$ mit $x_{i,j,s} = 1$.

- $T(i,j)$ enthält mindestens einen Eintrag $s \in Q \cup \Gamma$:

$$\phi_{\geq 1} = \bigvee_{s \in Q \cup \Gamma} x_{i,j,s}.$$

- $T(i,j)$ enthält höchstens einen Eintrag $s \in Q \cup \Gamma$:

$$\phi_{\leq 1} = \bigwedge_{s,t \in Q \cup \Gamma, s \neq t} \neg(x_{i,j,s} \wedge x_{i,j,t}).$$

- Liefert insgesamt $\phi_{Eintrag} = \bigwedge_{1 \leq i,j \leq n^k+1} (\phi_{\geq 1} \wedge \phi_{\leq 1})$.