Reelle Komplexität - Grundlagen II

Julian Bitterlich

TECHNISCHE UNIVERSITÄT DARMSTADT

Themenübersicht:

- Beziehungen zwischen den Komplexitätsklassen
  - Savitchs Theorem
- **coNP** und Charakterisierungen von **NP** und **coNP**
- Reduktion, Vollständigkeit, Härte
- **SAT** ist **NP**-vollständig
- **QSAT** ist **PSPACE**-vollständig

Beziehungen zwischen den Komplextitätsklassen

TECHNISCHE UNIVERSITÄT DARMSTADT

Wir wollen folgende Inklusionen zeigen:

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{P} \subseteq \mathbf{NP} \subseteq \mathbf{PSPACE} = \mathbf{NPSPACE} \subseteq \mathbf{EXP} \subseteq \mathbf{NEXP}$$

Dabei sind klar:

- $\mathbf{L} \subseteq \mathbf{NL}$✓
- $\mathbf{P} \subseteq \mathbf{NP}$✓
- $\mathbf{PSPACE} \subseteq \mathbf{NPSPACE}$✓
- $\mathbf{EXP} \subseteq \mathbf{NEXP}$✓

Es bleiben also noch:

- $\mathbf{NP} \subseteq \mathbf{PSPACE}$
- $\mathbf{NL} \subseteq \mathbf{P}$ und $\mathbf{NPSPACE} \subseteq \mathbf{EXP}$
- $\mathbf{NPSPACE} \subseteq \mathbf{PSPACE}$

$\mathbf{NTIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

Eine Funktion $f(n)$ heißt platzkonstruierbar/platzberechenbar, wenn es eine TM T gibt, so dass T, gegeben n in unärer Darstellung, $f(n)$ in unärer Darstellung berechnet. Dabei ist $T \in \mathbf{SPACE}(f(n))$.

kurz: $\overbrace{1 \ldots 1}^{n\text{-mal}} \xrightarrow{T} \overbrace{1 \ldots 1}^{f(n)\text{-mal}}$

Beispiel

$\log(n)$, n^k, $\sqrt{n}$, $n!$ sind platzberechenbar. Die Summe oder das Produkt zweier platzberechnbarer Funktionen ist wieder platzberechenbar.

Theorem

Sei $f(n)$ platzkonstruierbar, dann ist $\mathbf{NTIME}(f(n)) \subseteq \mathbf{SPACE}(f(n))$.

$$\Rightarrow \mathbf{NP} \subseteq \mathbf{PSPACE}\checkmark$$

$\mathbf{NSPACE}(f(n)) \subseteq \mathbf{TIME}(c^{\log(n)+f(n)})$

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

Eine Funktion $f(n)$ mit $f(n) \geq n$ heißt zeitkonstruierbar, wenn es eine TM T gibt, so dass T, gegeben n in unärer Darstellung, $f(n)$ in unärer Darstellung berechnet. Dabei ist $T \in \mathbf{TIME}(f(n))$.

Beispiel

n^k, $n!$, $n\log(n)$ sind zeitkonstruierbar.

Theorem

Sei $f(n)$ zeitkonstrierbar, dann ist $\mathbf{NSPACE}(f(n)) \subseteq \mathbf{TIME}(c^{\log(n)+f(n)})$.

Theorem

Sei $f(n)$ zeitkonstrierbar, dann ist $\mathbf{NSPACE}(f(n)) \subseteq \mathbf{TIME}(c^{\log(n)+f(n)})$.

Beweisskizze.

- Es gibt nur endlich viele Konfigurationen
- Die Konfigurationen zusammen mit den Übergangsrelationen bilden einen gerichteten Graphen
- Startknoten ist $C_0 = (s, \triangleright, x, \triangleright, \epsilon, \ldots, \triangleright, \epsilon)$
- Endknoten ist (O.B.d.A) $C = (q_{accept}, \triangleright, \epsilon, \triangleright, \epsilon, \ldots, \triangleright, \epsilon)$
- $\mathbf{REACHABILITY} \in \mathbf{TIME}(n^2)$

□

$$\Rightarrow \mathbf{NL} \subseteq \mathbf{P}\checkmark, \mathbf{NPSPACE} \subseteq \mathbf{EXP}\checkmark$$

Savitch's Theorem

TECHNISCHE UNIVERSITÄT DARMSTADT

Theorem (Savitch's Theorem)

$\mathbf{NSPACE}(f(n)) \subseteq \mathbf{SPACE}\left(f^2(n)\right)$ *für platzkonstruiebare Funktionen* $f(n) \geq \log(n)$.

Theorem

$\mathbf{REACHABILITY} \in \mathbf{SPACE}\left(\log^2(n)\right)$.

Beweisskizze.

- PATH(x, y, i) := „Es gibt einen Weg von x nach y, der höchstens 2^i lang ist“
- Es existiert ein Weg von x nach $y \Leftrightarrow$ PATH$(x, y, \log(n))$
- Wenn $i = 0$, kann PATH(x, y, i) an den Transitionen überprüft werden
- Wenn $i \geq 0$, dann PATH$(x, y, i) \Leftrightarrow \exists z(\text{PATH}(x, z, i-1) \wedge \text{PATH}(z, y, i-1))$
- Es müssen höchstens $\log(n)$ Tupel (x, y, i) der Länge $3\log(n)$ gespeichert werden

Savitch's Theorem

TECHNISCHE UNIVERSITÄT DARMSTADT

Theorem (Savitch's Theorem)

$\mathbf{NSPACE}\,(f(n)) \subseteq \mathbf{SPACE}\left(f^2(n)\right)$ *für Funktionen* $f(n) \geq n$.

$$\Rightarrow \mathbf{NPSPACE} \subseteq \mathbf{PSPACE}\checkmark$$

Was wir nicht machen...

TECHNISCHE UNIVERSITÄT DARMSTADT

Man kann zeigen, dass $\mathbf{P} \subsetneq \mathbf{EXP}$. Der Großteil der Forscher glaubt, dass

$$\mathbf{P} \subsetneq \mathbf{NP} \wedge \mathbf{NP} \subsetneq \mathbf{PSPACE} \wedge \mathbf{PSPACE} \subsetneq \mathbf{EXP}.$$

Aber man weiß nur

$$\mathbf{P} \subsetneq \mathbf{NP} \vee \mathbf{NP} \subsetneq \mathbf{PSPACE} \vee \mathbf{SPACE} \subsetneq \mathbf{EXP}.$$

Alternativer Characterisierung von $\mathbf{NP}$

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

Eine Sprache $L \subset \Sigma^*$ ist in $\mathbf{NP}$, wenn es ein $k \in \mathbb{N}$ und eine NTM M gibt, so dass L von M entschieden wird und $L \in \mathcal{O}(n^k)$ ist.

Theorem

Eine Sprache $L \subset \Sigma^*$ ist in $\mathbf{NP}$, genau dann wenn es ein Polynom $p : \mathbb{N} \to \mathbb{N}$ und eine TM M aus $\mathbf{P}$ (Prüfer genannt) gibt, so dass für jedes $x \in \Sigma^*$ gilt

$$x \in L \Leftrightarrow \exists u \in \Sigma^{p(|x|)} \text{ so dass } M(x, u) \text{ akzeptiert.} \tag{1}$$

Wenn $x \in L$ und $u \in \Sigma^{p(|x|)}$, so dass $M(x, u)$ akzeptiert, nennen wir u ein Zertifikat für x.

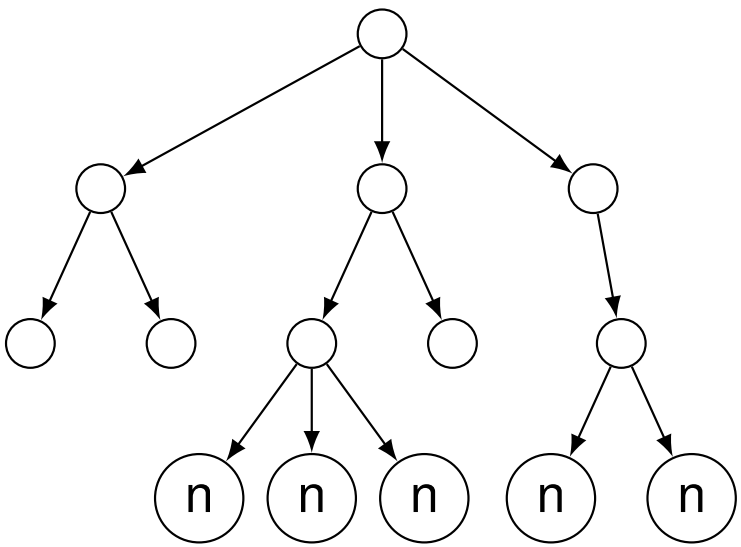
coNP

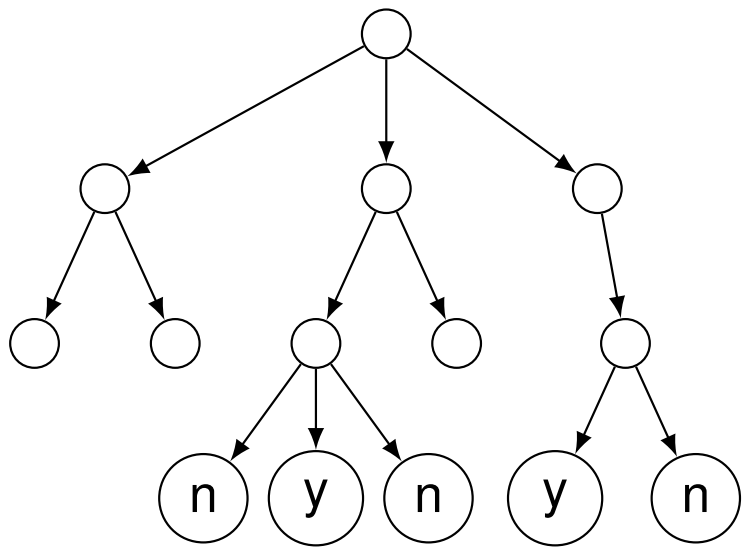
TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

Eine Sprache $L \subset \Sigma^*$ ist in **coNP**, wenn es $k \in \mathbb{N}$ und eine NTM M gibt, so dass L^c von M entschieden wird und $L \in \mathcal{O}(n^k)$ ist.

Kurz: $\mathbf{coNP} = \{L : L^c \in \mathbf{NP}\}$





Etwas mehr Gedanken zu $\mathbf{co}$

TECHNISCHE UNIVERSITÄT DARMSTADT

- Für jede Komplexitätsklasse $\mathcal{C}$, kann man diese Definition anwenden und erhält $\mathbf{co}\mathcal{C}$
- Ist $\mathcal{C}$ eine deterministische Komplexitätsklasse so ist $\mathbf{co}\mathcal{C} = \mathcal{C}$
- Sei M eine NTM, die die Sprache $L \subseteq \Sigma^*$ erkennt, und $x \in \Sigma^*$:
  - Es reicht *eine* Berechnung um $x \in L$ „semi zu entscheiden“
  - Man braucht *alle* Berechnungen um $x \notin L$ „semi zu entscheiden“
  - Bildet man aus M eine neue NTM $\overline{M}$, durch verstauchen von q_{accept} und q_{reject}, dann erkennt $\overline{M}$ nicht L^c sondern eine größere Sprache
- Im allgemeinen ist $\mathbf{co}\mathcal{C} \neq \mathcal{C}$
- Im allgemeinen ist $\mathbf{co}\mathcal{C} \neq \mathcal{C}^c$

Intermezzo: CNF-Formeln

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

Eine Formel $\phi(u_1, \ldots, u_n)$ ist in CNF (*Conjunctive Normal Form*), wenn sie von der Form ist

$$\bigwedge_i \left(\bigvee_j \nu_{i_j} \right)$$

ist, wobei ν_{i_j} entweder eine Variable u_k ist, oder eine negierte Variable $\neg u_k$ (Auch genannt Literal).
Mit anderen Worten: Eine Formel ist in CNF, wenn sie die Verundung von veroderten Literalen ist.
Die Subformeln $\bigvee_j \nu_{i_j}$ nennt man auch Klauseln.
Eine kCNF Formel ist eine CNF Formel dessen Klauseln höchstens k Literale haben.

Intermezzo: SAT und TAUT

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

SAT ist die Sprache aller erfüllbaren CNF-Formeln (in einer geeigneten Codierung).
z.B.

- $(a \vee b \vee c \vee d) \wedge (\neg a \vee b \vee e) \wedge (a \vee \neg c \vee \neg d) \wedge (\neg e) \wedge (c \vee \neg b) \in$ **SAT**

kSAT ist die Sprache aller erfüllbaren kCNF-Formeln.
z.B.

- $(a \vee b \vee c \vee d) \wedge (\neg a \vee b \vee e) \wedge (a \vee \neg c \vee \neg d) \wedge (\neg e) \wedge (c \vee \neg b) \notin$ **3SAT**
- $(x \vee y \vee \neg z) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg z) \wedge (\neg y \vee z) \wedge (\neg x \vee y) \wedge (x \vee z) \notin$ **3SAT**

Definition

TAUT ist die Sprache aller (immer) wahren CNF-Formeln.

Bemerke: $\phi \in$ **SAT** $\Leftrightarrow \neg\phi \notin$ **TAUT**.

Zurück zu NP und coNP

TECHNISCHE UNIVERSITÄT DARMSTADT

- **SAT** $\in$ **NP**
  Zertifikat ist eine erfüllende Belgung.
- **TAUT** $\in$ **coNP**
  Zertifikat, dass man nicht in **TAUT** ist, ist eine *nicht* erfüllende Belegung.
  Sallop formuliert: Damit $\phi \in$ **TAUT** gilt, muss ϕ jedem „Unzertifikat“ standhalten.

Theorem

Eine Sprache $L \subset \Sigma^*$ *ist in* **coNP**, *genau dann wenn es ein Polynom* $p : \mathbb{N} \to \mathbb{N}$ *und eine* TM M *aus* **P** *gibt, so dass für jedes* $x \in \Sigma^*$ *gilt*

$$x \in L \Leftrightarrow \forall u \in \Sigma^{p(|x|)} \quad M(x, u) \text{ akzeptiert.} \tag{2}$$

Folgerungen und Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Da $\mathbf{PSPACE} = \mathbf{coPSPACE}$ und $\mathbf{NPSPACE} = \mathbf{PSPACE}$ gilt $\mathbf{NPSPACE} = \mathbf{coNPSPACE}$
- $\mathbf{P} \subseteq \mathbf{NP} \cap \mathbf{coNP}$
- $\mathbf{NP} \cup \mathbf{coNP} \subseteq \mathbf{PSPACE}$
- Wenn $\mathbf{P} = \mathbf{NP}$, dann ist auch $\mathbf{coNP} = \mathbf{NP}$
- $\mathbf{NSPACE}(f(n)) = \mathbf{coNSPACE}(f(n))$ für platzkonstruierbare Funktionen $f(n) \geq \log(n)$
- Die Polynomielle Hirachie

Reduktion

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

Eine Sprache K ist polynomiell reduzierbar auf die Sprache L, wenn es eine polynomielle TM T gibt mit Ein- und Ausgabe, so dass

$$x \in K \Leftrightarrow T(x) \in L. \tag{3}$$

Man schreibt $K \leq_p L$.
Eine Komplexitätsklasse $\mathcal{C}$ heißt abgeschlossen under (polynomieller) Redunktion, wenn für alle Sprachen $K, L \subseteq \Sigma^*$

$$K \leq_p L \wedge L \in \mathcal{C} \Rightarrow K \in \mathcal{C} \tag{4}$$

gilt.
Eine Sprache $L \subseteq \Sigma^*$ heißt $\mathcal{C}$-hart, wenn sich alle Sprachen $K \in \mathcal{C}$ auf L reduzieren lassen ($K \leq_p L$).
Eine Sprache $L \subseteq \Sigma^*$ heißt $\mathcal{C}$-vollständig, wenn sie $\mathcal{C}$-hart ist und in $\mathcal{C}$ liegt.

Mehr zu Reduktion

TECHNISCHE UNIVERSITÄT DARMSTADT

Bemerkung

- $\mathbf{P}, \mathbf{NP}, \mathbf{coNP}, \mathbf{PSPACE}, \mathbf{EXP}, \mathbf{NEXP}$ sind abgeschlossen unter Reduktion
- Reduktion ist für $\mathbf{P}$ nicht besonders sinnvoll
- Reduktion ist eine Äquivalenzrealtion (insbesondere Transitiv)
- Es gibt noch andere von Reduktion, z.B. Reduktion in logarithmischen Platz (Dann auch sinnvoll für $\mathbf{P}$ und $\mathbf{NL}$)

Theorem

Seien $\mathcal{C}, \mathcal{D}$ zwei Komplexitätsklassen und $L \subseteq \Sigma^*$ eine Sprache. Ist $\mathcal{C}$ abgeschlossen unter Reduktion und L $\mathcal{D}$-vollständig, dann ist $\mathcal{D} \subseteq \mathcal{C}$.

SAT ist NP-vollständig

TECHNISCHE UNIVERSITÄT DARMSTADT

Beweis.

- O.B.d.A sei die Maschine M eine 2Band-TM, *ignorant* und $\Sigma = \{0, 1\}$
- Wegen (1) gibt es eine TM M, so dass für alle $x \in \Sigma^*$, $x \in L \Leftrightarrow M(x, u) = 1$ für ein $u \in \Sigma^{p(|x|)}$, wobei p ein Polynom ist.
- Konstruktion einer Reduktion $x \to \phi_x$, so dass, $\phi_x \in$ **SAT** $\Leftrightarrow \exists u \in \Sigma^{p(|x|)}$ mit $M(x, u) = 1$
- Ein Schnappschuss ist ein Tripel $\langle a, b, q \rangle \in \Gamma \times \Gamma \times Q$. Die Länge hängt von Γ und Q ab.
- zur Zeit i hängt der Schnappschuss z_i nur von $z_{i-1}, y_{inputpos(i)}, z_{prev(i)}$
- Es gibt eine boolsche Funktion F, so dass $F(z_{i-1}, z_{prev(i)}, y_{inputpos(i)}) = z_i$

SAT ist NP-vollständig

TECHNISCHE UNIVERSITÄT DARMSTADT

Beweis.

Ziel: Bilde Formel ϕ_x, so dass eine Erfüllenden Belegung genau die Form hat $z_1, \ldots, z_{T(n)}$, mit

- Die ersten n bits von y sind gleich x
- $z_1 = \langle \triangleright, \triangleright, q_{start} \rangle$
- Für jedes $i \in \{2, \ldots, T(n)\}$, $z_i = F(z_{i-1}, z_{inputpos(i)}, z_{prev(i)})$
- $z_{T(n)} = \langle \triangleright, \triangleright, q_{accpept} \rangle$

Dies kann in polynomieller Zeit geschehen. □

Mehr vollständigkeit

TECHNISCHE UNIVERSITÄT DARMSTADT

- TAUT ist $\mathbf{coNP}$-vollständig
- INDESET ist $\mathbf{NP}$-vollständig
- 3SAT ist $\mathbf{NP}$-vollständig
- HAMPATH ist $\mathbf{NP}$-vollständig
- CLIQUE ist $\mathbf{NP}$-vollständig
- TSP ist $\mathbf{NP}$-vollständig

  ⋮
- Lander's Theorem: Wenn $\mathbf{P} \neq \mathbf{NP}$, dann gibt es eine Sprache $L \in \mathbf{NP} \setminus \mathbf{P}$ die nicht $\mathbf{NP}$-vollständig ist

QSAT

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition

QSAT ist die Sprache aller wahren *quantifizierten* boolschen Formeln in pränex Normalform $\psi = Q_1 \ldots Q_n \phi(x_1, \ldots, x_n)$, wobei Q_i einet der beiden Quantoren $\exists$ oder $\forall$ ist.

Theorem

QSAT *ist in* **PSPACE**.

QSAT ist PSPACE-vollständig

TECHNISCHE UNIVERSITÄT DARMSTADT

Beweis.

Sei $L \in \mathbf{PSPACE}$ und M die entsprechende TM

- O.B.d.A $\Sigma = \{0, 1\}$.
- Eine Konfiguration braucht $m = \mathcal{O}(n)$ Platz.
- der Konfigurationsbaum hat höchstens 2^m Elemente.
- Es gibt eine boolsche Formel $\phi_{m,x}$, so dass für alle $C, C' \in \{0, 1\}^m$, $\phi_{M,x}(C, C') = 1$ genau dann wenn C und C' zwei aufeinander folgenden Konfigutationen sind.
- Konstruieren $\psi_i \in \mathbf{QSAT}$, so dass $\psi_i(C, C')$ genau dann wahr ist, wenn es im Konfigurationsgraph einen Weg von C nach C' kürzer gleich 2^i gibt.
- $\psi_m(C_{start}, C_{accpet})$ ist die gesuchte Formel.

Reduktion als relatver Beweis, dass ein Problem schwer ist

TECHNISCHE UNIVERSITÄT DARMSTADT

- $\textsc{Sat} \in \mathbf{P} \Leftrightarrow \mathbf{P} = \mathbf{NP}$
- Obiges gilt für alle $\mathbf{NP}$-vollständigen Probleme. Diese liegen am unwahrscheinlichsten in $\mathbf{P}$
- $\textsc{QSat} \in \mathbf{NP} \Leftrightarrow \mathbf{PSPACE} = \mathbf{NP}$