Berechenbarkeit und Komplexität
13. Vorlesung

Prof. Dr. Dietrich Kuske

FG Automaten und Logik, TU Ilmenau

Sommersemester 2023

Ziel dieser Vorlesung

Wer SAT „schnell“ lösen kann, der kann alle Probleme in NP (von denen es sehr viele gibt) „schnell“ lösen.

Typische Probleme, 2. Versuch

Satz

SAT $\in$ NP

Beweis: Sei φ eine aussagenlogische Formel, in der die atomaren Formeln $x_1, \ldots, x_n$ vorkommen.

Eine nichtdeterministische Turingmaschine „rät“ nun in einer ersten Phase eine Belegung $\mathcal{B}: \{x_1, \ldots, x_n\} \to \{0, 1\}$:

- Im ersten Schritt schreibt sie x_1 auf das Band und danach nichtdeterministisch 0 oder 1.
- Im zweiten Schritt schreibt sie x_2 auf das Band und danach nichtdeterministisch 0 oder 1.
- Im dritten Schritt ...

Nach n Schritten steht ein Wort der Form $x_1 b_1 x_2 b_2 \cdots x_n b_n$ mit $b_1, \ldots, b_n \in \{0, 1\}$ auf dem Band.

Dieses Wort kodiert die Belegung $\mathcal{B}$ mit $\mathcal{B}(x_i) = b_i$ für $1 \leq i \leq n$.

In einer zweiten Phase kann die Turingmaschine nun den Wert $\mathcal{B}(\varphi)$ deterministisch ausrechnen, indem die Formel φ einmal von links nach rechts durchlaufen, jede atomare Formel x_i durch den Wert $\mathcal{B}(x_i) = b_i$ aus dem in der ersten Phase erzeugten „Belegungswort“ ersetzt, und dann die Formel „ausgerechnet“ wird.

Die Maschine geht in eine akzeptierende Haltekonfiguration, wenn sie bei der Auswertung 1 erhält, sonst geht sie in eine nicht akzeptierende Haltekonfiguration.

Dies benötigt höchstens $|\varphi|^3$ Schritte, die TM hält also auf jeden Fall nach $O\big(|\varphi|^3\big)$ Schritten.

Außerdem gibt es genau dann eine akzeptierende Berechnung, wenn φ erfüllbar ist. □

Analog kann man zeigen:

Satz

$HC, 3C \in NP$

Beweis: rate in polynomieller Zeit eine Lösung und überprüfe in polynomieller Zeit, daß es sich tatsächlich um eine solche handelt. □

Die „typischen Probleme“ SAT, HC, 3C sind also nicht nur in PSPACE, sondern sogar in $NP \subseteq PSPACE$ (ob hier Gleichheit gilt, ist nicht bekannt).

Aber liegen sie auch in $P \subseteq NP$, d.h. sind sie durch eine deterministische TM in polynomieller Zeit lösbar?

Polynomialzeit-Reduktionen

Erinnerung: Reduktionen erlauben es, Probleme bzgl. der „Schwere der Lösbarkeit“ zu vergleichen.

Allerdings sind sie für entscheidbare Probleme nicht sehr aussagekräftig:

Fakt

Seien $A, B \subseteq \Sigma^*$ entscheidbare Sprachen mit $\emptyset \neq B \neq \Sigma^*$. Dann gilt $A \leq B$.

Beweis: Übung 4.4(b) □

Wir werden jetzt Polynomialzeit-Reduktionen betrachten, die einen genaueren Vergleich der „Schwere der Lösbarkeit“ ermöglichen.

Definition

1. Eine Funktion $f: \Sigma^* \to \Gamma^*$ ist polynomial berechenbar, falls eine Turingmaschine M und ein Polynom $p(n) \in \text{Poly}$ existieren, so daß für alle $w \in \Sigma^*$ gilt:

   Wenn M mit der Eingabe w gestartet wird, hält M nach höchstens $p\big(|w|\big)$ vielen Schritten mit der Ausgabe $f(w)$ an.
2. Eine Sprache $A \subseteq \Sigma^*$ ist polynomial reduzierbar auf eine Sprache $B \subseteq \Gamma^*$ (kurz $A \leq_P B$), falls eine polynomial berechenbare Funktion $f: \Sigma^* \to \Gamma^*$ existiert mit

$$\forall w \in \Sigma^* : \big(w \in A \iff f(w) \in B \big).$$

Lemma (vgl. Folie 7.21)

Wenn $A \leq_P B$ und $B \in \mathrm{P}$ (bzw. $B \in \mathrm{NP}$), dann gilt $A \in \mathrm{P}$ (bzw. $A \in \mathrm{NP}$).

Beweis:

Sei zunächst $A \leq_P B$ und $B \in \mathrm{P}$.

Dann existieren Polynome $p(n)$ und $q(n)$ sowie Turingmaschinen M und N mit folgenden Eigenschaften:

- M berechnet aus einer Eingabe $w \in \Sigma^*$ in Zeit $p\big(|w|\big)$ ein Wort $f(w)$, so daß gilt $w \in A \iff f(w) \in B$.

  Beachte: Da die Maschine M in $p\big(|w|\big)$ Schritten nur eine Ausgabe der Länge höchstens $p\big(|w|\big) + |w|$ erzeugen kann, gilt $\big|f(w)\big| \leq p\big(|w|\big) + |w|$.

- N akzeptiert die Sprache B in Zeit $q(n)$.

Eine Turingmaschine für die Sprache A arbeitet dann bei einer Eingabe w wie folgt:

1. Berechne $f(w)$ (Zeitbedarf: $p\big(|w|\big)$).
2. Simuliere die Maschine N auf $f(w)$ (Zeitbedarf: $q\big(|f(w)|\big)$).

Der gesamte Zeitbedarf ist also

$$p\big(|w|\big) + q\big(|f(w)|\big) \leq p\big(|w|\big) + q\Big(p\big(|w|\big) + |w|\Big),$$

was wieder ein Polynom ist.

Die Aussage für die Klasse NP kann genauso bewiesen werden. □

NP-Vollständigkeit

Definition

Eine Sprache B ist NP-hart, falls für alle $A \in \text{NP}$ gilt: $A \leq_P B$
(A ist mindestens so schwer wie jedes Problem in NP).

Eine Sprache ist NP-vollständig, falls sie zu NP gehört und NP-hart ist.

Intuition:

- NP-vollständige Sprachen sind die schwierigsten Sprachen in NP.
- Ist B NP-vollständig, so gilt $\text{NP} = \left\{ A \text{ Sprache} \,\middle|\, A \leq_P B \right\}$

Noch wissen wir gar nicht, ob es überhaupt NP-vollständige Probleme gibt.
Dies werden wir bald zeigen.

Zunächst aber noch ein einfaches Resultat:

Lemma

Wenn B NP-vollständig ist, dann gilt: $\mathrm{P} = \mathrm{NP} \iff B \in \mathrm{P}$.

Beweis:

„$\Rightarrow$“: Sei $\mathrm{P} = \mathrm{NP}$.
Da B NP-vollständig ist, folgt $B \in \mathrm{NP} = \mathrm{P}$.

„$\Leftarrow$“: Sei $B \in \mathrm{P}$ und sei $A \in \mathrm{NP}$ beliebig.
Da B NP-vollständig ist, folgt $A \leq_P B \in \mathrm{P}$.
Das Lemma auf Folie 13.8 impliziert $A \in \mathrm{P}$.
Also gilt $\mathrm{NP} \subseteq \mathrm{P}$ und damit $\mathrm{NP} = \mathrm{P}$. □

Satz (Stephen Cook 1939-, 1971 & Leonid Levin 1948-, 1973)

SAT ist NP-vollständig.

Dieser Satz besagt insbesondere, daß es ein natürliches NP-vollständiges Problem gibt.

Beweisstrategie: Sei $A \subseteq \Sigma^*$ in NP

$\Rightarrow \exists$ Polynom p und $p(n)$-zeitbeschränkte NTM M, die A akzeptiert

o.E. habe jede Berechnung bei Eingabe von w genau die Länge $p\big(|w|\big)$

Aus $w \in \Sigma^*$ konstruieren wir aussagenlogische Formel φ_w mit

$w \in A$ gdw. es gibt akzeptierende Berechnung
$C_0 \vdash_M C_1 \vdash_M \cdots \vdash_M C_{p(|w|)}$ bei Eingabe von w

gdw. φ_w ist erfüllbar, d.h. $\varphi_w \in$ SAT

da φ_w polynomial berechnet werden kann, ist Abbildung $w \mapsto \varphi_w$ Polynomialzeitreduktion von A auf SAT, d.h. $A \leq_P$ SAT

Seien $\Gamma = \{a_0, a_1, \ldots, a_\ell\}$ Bandalphabet, $Z = \{z_0, \ldots, z_k\}$ Menge der Zustände der NTM M und $w = b_1 b_2 \ldots b_n \in \Sigma^*$.

wir verwenden die folgenden atomaren Formeln

Atomformel	Indizes	intendierte Bedeutung
$\text{zust}_{t,z}$	$0 \leq t \leq p(n)$ $z \in Z$	nach t Schritten befindet sich TM im Zustand z
$\text{pos}_{t,i}$	$0 \leq t \leq p(n)$ $-p(n) \leq i \leq p(n)$	nach t Schritten befindet sich Kopf auf Position i
$\text{band}_{t,i,a}$	$0 \leq t \leq p(n)$ $-p(n) \leq i \leq p(n)$ $a \in \Gamma$	nach t Schritten steht in Zelle i der Buchstabe a

Ziel: Formel $\varphi_w = \bigwedge_{1 \leq i \leq 5} \varphi_w^i$, so daß für alle Belegungen $\mathcal{B}$ gilt:

$$\mathcal{B}(\varphi_w) = 1 \iff \mathcal{B} \text{ „kodiert“ akzeptierende Berechnung von } M \text{ bei Eingabe von } w.$$

Lemma (= Hilfssatz)

Aus $n \in \mathbb{N}$ kann eine aussagenlogische Formel $\gamma_n(x_0, \ldots, x_n)$ in Zeit $O(n^3)$ berechnet werden, so daß

- $|\gamma_n| \in O(n^3)$ und
- für alle Belegungen $\mathcal{B}$ gilt:

$$\mathcal{B}(\gamma_n) = 1 \text{ gdw. } \mathcal{B}(x_i) = 1 \text{ für genau ein } i \text{ mit } 0 \leq i \leq n$$

Beweis:

$$\gamma_n = \bigvee_{0 \leq i \leq n} x_i \wedge \bigwedge_{0 \leq i < j \leq n} \neg(x_i \wedge x_j)$$

erste Teilformel: wenigstens ein x_i ist wahr
zweite Teilformel: es gibt nicht zwei verschiedene atomare Formeln, die wahr sind

also: $\mathcal{B}(\gamma_n) = 1$ gdw. $\mathcal{B}(x_i) = 1$ für genau ein i

$|\gamma_n| \leq c \cdot (n+1)^2 + d \cdot (n+1)^3 \in O(n^3)$ für gewisse c und d □

φ_w^1 ist die folgende Formel

$$\bigwedge_{0 \le t \le p(n)} \left[\begin{array}{ll} & \gamma_k(\mathrm{zust}_{t,z_0}, \mathrm{zust}_{t,z_1}, \ldots \mathrm{zust}_{t,z_k}) \\ \wedge & \gamma_{2p(n)}(\mathrm{pos}_{t,-p(n)}, \mathrm{pos}_{t,-p(n)+1}, \ldots \mathrm{pos}_{t,p(n)}) \\ \wedge & \bigwedge_{-p(n) \le i \le p(n)} \gamma_\ell(\mathrm{band}_{t,i,a_0}, \mathrm{band}_{t,i,a_1}, \ldots \mathrm{band}_{t,i,a_\ell}) \end{array} \right]$$

Sie sagt aus: zu jedem Zeitpunkt t mit $0 \le t \le p(n)$ gilt:

- die TM ist in genau einem Zustand (erste Zeile)
- der Kopf ist an genau einer Position (zweite Zeile)
- an jeder Position zwischen $-p(n)$ und $p(n)$ steht genau ein Symbol aus Γ (letzte Zeile)

φ^2_w ist die folgende Formel

$$\text{zust}_{0,z_0} \wedge \text{pos}_{0,0} \wedge \bigwedge_{0 \le i < n} \text{band}_{0,i,b_{i+1}} \wedge \bigwedge_{\substack{-p(n) \le i < 0 \\ \text{oder } n \le i \le p(n)}} \text{band}_{0,i,\square} .$$

Sie sagt aus: zum Zeitpunkt 0 gilt:

- die TM ist im Initialzustand z_0,
- der Kopf befindet sich auf der Position 0,
- auf den Positionen $0, 1, \ldots, n-1$ steht das Wort $w = b_1 b_2 \ldots b_n$ und
- auf restlichen Positionen zwischen $-p(n)$ und $p(n)$ steht $\square$.

D.h. die Formel sagt aus, daß TM in Anfangskonfiguration mit Eingabe w startet.

φ_w^3 ist die folgende Formel

$$\bigwedge_{(a)} \left[\begin{array}{l} (\text{zust}_{t,z} \wedge \text{pos}_{t,i} \wedge \text{band}_{t,i,a}) \\ \qquad \rightarrow \bigvee_{(b)} (\text{zust}_{t+1,z'} \wedge \text{pos}_{t+1,i+y} \wedge \text{band}_{t+1,i,a'}) \end{array} \right].$$

Hierbei steht

(a) für $0 \leq t < p(n)$, $z \in Z$, $-p(n) \leq i \leq p(n)$, $a \in \Gamma$

(b) für $(z', a', M) \in \delta(z, a)$, $y = \begin{cases} -1 & \text{falls } M = L \\ 0 & \text{falls } M = N \\ 1 & \text{falls } M = R \end{cases}$

Sie sagt aus: wenn zum Zeitpunkt t die Maschine im Zustand z ist und das Zeichen a auf Position i liest, so existiert eine Anweisung $(z', a', M) \in \delta(a, z)$, so daß Maschine zum Zeitpunkt $t + 1$ im Zustand z' ist, an Position i das Symbol a' stehtund der Kopf sich entsprechend bewegt hat.

φ_w^4 ist die folgende Formel

$$\bigwedge_{(a)} \Big((\neg \mathrm{pos}_{t,i} \wedge \mathrm{band}_{t,i,a}) \rightarrow \mathrm{band}_{t+1,i,a} \Big)$$

wobei (a) für $0 \leq t < p(n)$, $-p(n) \leq i \leq p(n)$, $a \in \Gamma$ steht.

Sie sagt aus: wenn zum Zeitpunkt t der Kopf nicht an Position i steht, so wird dieses Symbol im nächsten Zeitpunkt nicht geändert.

φ_w^5 ist die folgende Formel

$$\bigvee_{(a)} \left(\text{pos}_{p(n),i} \wedge \text{zust}_{p(n),z} \wedge \text{band}_{p(n),i,a} \wedge \bigwedge_{-p(n) \leq j < i} \text{band}_{p(n),j,\square} \right).$$

wobei (a) für $-p(n) \leq i \leq p(n)$, $z \in E$, $a \in \Gamma$ mit $\delta(z,a) \subseteq \{(z,a,N)\}$ steht. Sie sagt aus, daß sich die TM zum Zeitpunkt $p(n)$ in einer akzeptierenden Haltekonfiguration befindet.

Damit ist die Konstruktion von $\varphi_w = \varphi_w^1 \wedge \varphi_w^2 \wedge \varphi_w^3 \wedge \varphi_w^4 \wedge \varphi_w^5$ abgeschlossen.

Noch zu zeigen:

- $w \in A$ gdw. φ_w erfüllbar
- Die Abbildung $w \mapsto \varphi_w$ ist polynomial berechenbar.

z.z.: $w \in A \implies \varphi_w$ erfüllbar

Sei $w \in A$. Dann existiert akzeptierende Berechnung von M bei Eingabe von w, diese hat die Länge $p(n)$. Da sich Kopf höchstens $p(n)$ viele Schritte bewegen kann, sind in jeder Konfiguration höchstens die Positionen $-p(n)$ bis $p(n)$ beschrieben. Mit in Tabelle auf Folien 13.13 angegebener Bedeutung der atomaren Formeln wird jede der Teilformeln φ_w^i wahr, also auch φ_w.

z.z.: $w \in A \iff \varphi_w$ erfüllbar

Sei nun $\mathcal{B}$ Belegung der Variablen mit $\mathcal{B}(\varphi_w) = 1$

- da $\mathcal{B}(\varphi_w^1) = 1$, bestimmen die Werte der atomaren Formeln $\text{zust}_{t,q}$, $\text{pos}_{t,i}$ und $\text{band}_{t,i,a}$ für jeden Zeitpunkt $0 \leq t \leq p(n)$ eine Konfiguration C_t
- da $\mathcal{B}(\varphi_w^2) = 1$, ist C_0 die Anfangskonfiguration bei Eingabe von w
- wegen $\mathcal{B}(\varphi_w^3 \wedge \varphi_w^4) = 1$ gilt $C_t \vdash_M C_{t+1}$ für alle $0 \leq i < p(n)$
- wegen $\mathcal{B}(\varphi_w^5) = 1$ ist $C_{p(n)}$ akzeptierende Konfiguration,

also wird w von M akzeptiert, d.h. $w \in L(M) = A$.

z.z.: φ_w kann in Polynomialzeit aus w berechnet werden

Länge von φ_w: es gibt Konstanten $c_i \geq 1$ mit

$$|\varphi_w^i| \leq c_i \cdot p\big(|w|\big)^4$$

für alle $i \in \{1,2,3,4,5\}$, also $|\varphi_w| \in O\Big(p\big(|w|\big)^4\Big)$.

aufgrund der einfachen Struktur von φ_w wird nur Zeit $|\varphi_w|$ benötigt, um diese Formel auszurechnen. □

Zusammenfassung 13. Vorlesung

in dieser Vorlesung neu

- Die von uns betrachteten typischen Problem sind in NP.
- Ein Problem ist NP-vollständig, wenn es zu NP gehört und sich jedes Problem aus NP in polynomieller Zeit darauf reduzieren läßt, es also „eines der schwersten Probleme in NP ist“.
- Satz von Cook-Levin: SAT is NP-vollständig.

Wer also SAT „schnell“ lösen kann, der kann alle Probleme in NP (von denen es sehr viele gibt) „schnell“ lösen.

kommende Vorlesung

- weitere natürliche NP-vollständige Probleme