Theorem 6.13

$P \subseteq P/poly$

Beweisskizze:

Ähnlich wie der Beweis des Satzes von Cook und Levin (NP-Vollständigkeit von SAT):

Sei $L \in P$. Dann gibt es ein $c \geq 1$ und ein $T(n) := n^c$ - zeitbeschränkte stereotype (engl.: oblivious) det. 1-Band TM M, die L entscheidet (vgl. Bemerkung 1.10)

D.h.: Die Kopfposition im i-ten Berechnungsschritt hängt nur von der Länge der Eingabe ab, aber nicht von der konkreten Eingabe selbst.

Wir können außerdem o.B.d.A. annehmen, dass M in den ersten $n = |x|$ Berechnungsschritten im Zustand q_0 einmal von links nach rechts über das Band läuft, ohne dessen Beschriftung zu verändern. (*)

Zur Konstruktion des SKes C_n betrachten wir für jede Eingabe $x \in \{0,1\}^n$ die Folge

$$z_0, z_1, \dots, z_n, z_{n+1}, \dots, z_{T(n)}$$

von sog. Schnappschüssen von M. D.h.: z_i gibt den aktuellen Zustand sowie das an der aktuellen Kopfposition gelesene Symbol an. Insbes. gilt wegen (*):

$z_0 = (q_0, \triangleright)$ und $z_i = (q_0, x_i)$ f.a. $i \in \{1, \dots, n\}$.

Außerdem gilt f.a. $i \in \{n+1, \dots, T(n)\}$:

(**): Durch die Überführungsfunktion δ von M ist

z_i eindeutig festgelegt durch z_{i-1} und $z_{prev(i)}$, wobei prev(i) der letzte Zeitpunkt vor i ist, zu dem der Kopf auf derselben Bandposition wie zum Zeitpunkt i war (bzw $z_{prev(i)} := \perp$, falls i der erste Zeitpunkt ist, an dem diese Bandposition betreten wird)

Da M stereotyp ist, hängt prev(i) nicht von der Eingabe x ab, sondern nur von i und $n := |x|$. (***)

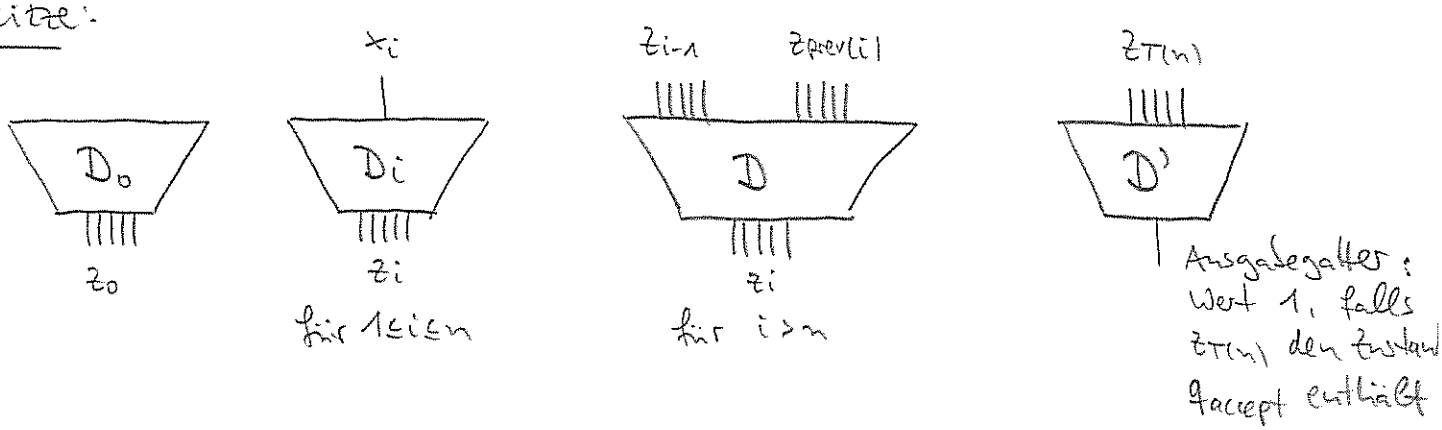
Jeden Schnappschuss können wir durch eine konstante Anzahl k von Bits repräsentieren (k hängt nur von der Zustandsmenge Q und dem Bandalphabet Γ von M ab, aber nicht von n oder x).

Unser SK C_n enthält für jeden der Schnappschüsse $z_0, z_1, \ldots, z_{T(n)}$ k Gatter, deren Wert bei Eingabe x den Schnappschuss repräsentieren.

- $z_0 = (q_0, \triangleright)$ kann durch einen SK D_0 berechnet werden, der aus k 0- bzw 1-Gattern besteht
- Für $i \in \{1, \ldots, n\}$ kann $z_i = (q_0, x_i)$ durch einen SK D_i berechnet werden, der aus dem Eingabegatter x_i sowie weiteren 0- bzw 1-Gattern besteht
- Wegen (**) gibt es einen SK D konstanter Größe, der für jedes $i > n$ bei Eingabe von z_{i-1} und $z_{prev(i)}$ den Schnappschuss z_i berechnet

- Außerdem gibt es einen SK D' konstanter Größe, der an seinem Ausgabegatter angibt, ob der Schnappschuss $z_{T(n)}$ den akzeptierenden Zustand q_{accept} enthält.

Skizze:



Der SK C_n wird aus den Sken $D_0, D_1, \dots, D_n, D'$ und $(T(n)-n)$ Sken der Form D zusammengesetzt. Wegen (***) hängt die "Verkabelung" von C_n nur von n ab (und nicht von der konkreten Eingabe x).

Insgesamt gilt:

- $|C_n| = O(T(n)) = O(n^c)$ und
- f.a. $x \in \{0,1\}^n$ ist

  $C_n(x) = 1 \iff M$ akzeptiert $x \iff x \in L$

Somit ist $(C_n)_{n \ge 0}$ eine SK-Familie polynomieller Größe, die L berechnet. Dh: $L \in P/poly$.

□

Bemerkung 6.14

Der im Beweis von Theorem 6.13 konstruierte SK C_n kann sogar von einer TM konstruiert werden — und zwar in Zeit poly(n) und auf Platz $O(\log n)$ (im Sinne von write-once logspace-Berechnungen).

Die stereotype TM M kann nämlich so gewählt werden, dass die Kopfposition zum Zeitpunkt i bei Eingabe eines Worts der Länge n auf Platz $O(\log n)$ berechnet werden kann.

Mit dieser Beweismethode lassen sich einige weitere Sätze zeigen — u.a. Folgendes:

Folgerung 6.15

(a) Das Problem

$$\text{CIRCUIT-EVAL} := \{ \langle C, x \rangle : C \text{ ist ein SK mit } n \text{ Eingabegattern}, n \in \mathbb{N}, x \in \{0,1\}^n, C(x) = 1 \}$$

ist P-vollständig im folgenden Sinn:

Ein Problem $L' \subseteq \{0,1\}^*$ ist P-vollständig, falls gilt:

- $L' \in P$ und
- L' ist P-hart, d.h. f.a. $L \in P$ gilt: $L \leq_e L'$

(Zur Erinnerung: $\leq_e$ bezeichnet logspace-Reduzierbarkeit)

(b) Das Problem

$$\text{CIRCUIT-SAT} := \{ C : C \text{ ist ein SK mit } n \text{ Eingabegattern } n \in \mathbb{N}, \text{ es gibt ein } x \in \{0,1\}^n \text{ s.d. } C(x) = 1 \}$$

ist NP-vollständig.

Beweis:
Man sieht leicht, dass CIRCUIT-EVAL in P und CIRCUIT-SAT in NP liegt.

(a) Die P-Härte von CIRCUIT-EVAL lässt sich aus dem Beweis von Theorem 6.13 und Bemerkung 6.14 folgern:

Für $L \in P$ erhalten wir $L \leq_e$ CIRCUIT-EVAL durch die Reduktion f, die ein $x \in \{0,1\}^*$ abbildet auf $f(x) := \langle C_{|x|}, x \rangle$, wobei $C_{|x|}$ der im Beweis von Theorem 6.13 konstruierte SK C_n für $n := |x|$ ist.
Es gilt: $x \in L \iff C_n(x) = 1 \iff \underbrace{\langle C_{|x|}, x \rangle}_{= f(x)} \in$ CIRCUIT-EVAL

Gemäß Bemerkung 6.14 ist f write-once logspace-berechenbar.

(b) Die NP-Härte von CIRCUIT-SAT erhalten wir wie folgt:
Sei $L \in NP$. Gemäß Definition 2.1 gibt es ein Polynom p und eine det. Polynomialzeit-TM M, s.d. f.a. $x \in \{0,1\}^*$ gilt: $x \in L \iff$ ex. $u \in \{0,1\}^{p(|x|)}$ s.d. $M(\langle x,u \rangle) = 1$.
Sei C_m der im Beweis von Theorem 6.13 für M konstruierte SK, mit $m = |\langle x,u \rangle|$, für den gilt:
$C_m(\langle x,u \rangle) = 1 \iff M(\langle x,u \rangle) = 1$.

Aus C_m und x lässt sich leicht ein SK C_x konstruieren, für den gilt:

$C_x \in$ CIRCUIT-SAT $\iff$ es $u \in \{0,1\}^{p(|x|)}$: $C_m(\langle x, u\rangle) = 1$ $\iff$ $x \in L$

(Details: Übung!)

Insgesamt liefert die Abbildung f mit $f(x) := C_x$ eine Polynomialzeit-Reduktion (sogar eine logspace-Reduktion) von L auf CIRCUIT-SAT. □

Theorem 6.13 besagt, dass $P \subseteq P/poly$ ist.

Frage: Gilt auch: $NP \subseteq P/poly$?

Antwort: Nein — es sei denn, die Polynomialzeit-Hierarchie kollabiert. Genauer:

Theorem 6.16 (Satz von Karp und Lipton, 1980)

Falls $NP \subseteq P/poly$, so ist $PH = \Sigma_2^P$

Beweis:

Wegen Satz 5.4 genügt es zu zeigen, dass $\Pi_2^P \subseteq \Sigma_2^P$.

Dazu genügt es, zu zeigen, dass das Π_2^P-vollständige Problem Π_2-SAT (vgl. Beobachtung 5.8) in Σ_2^P liegt.

Falls $NP \subseteq P/poly$, so gilt insbesondere: SAT $\in P/poly$.

D.h.: Es gibt eine SK-Familie $(C_n)_{n \in \mathbb{N}}$ polynomieller Größe, s.d. für C_n bei Eingabe einer aussagenlogischen Formel φ der Länge n gilt:

$C_n(\varphi) = 1$ $\iff$ φ ist erfüllbar.

Unter Verwendung des im Beweis von Satz 2.13 genutzten Algorithmus lässt sich aus $(C_n)_{n\in\mathbb{N}}$ eine SK-Familie $(C'_n)_{n\in\mathbb{N}}$ polynomieller Größe konstruieren, für die folgendes gilt:

⊛: $C'_n(\varphi)$ gibt an, ob φ erfüllbar ist. Und es gibt Gatter $z_1 \dots z_n$ in C'_n, s.d. gilt: Falls $C'_n(\varphi) = 1$, so gibt der Wert der Gatter $z_1 \dots z_n$ eine erfüllende Belegung von φ an.

Da $(C'_n)_{n\in\mathbb{N}}$ polynomielle Größe hat, gibt es ein Polynom $p: \mathbb{N} \to \mathbb{N}$ s.d. C'_n durch einen Bitstring der Länge $p(n)$ beschrieben werden kann.

Wir wollen nun ⊛ nutzen, um zu zeigen, dass $\Pi_2\text{-SAT} \in \Sigma_2^P$:

Zur Erinnerung:

$\Pi_2\text{-SAT} = \{ \Phi : \Phi$ ist eine wahre QBF der Form $\forall \bar{u}\, \exists \bar{v}\, \varphi$, wobei $\bar{u}$ und $\bar{v}$ Listen von Variablen sind und φ eine aussagenlogische Formel über den Variablen $\bar{u}, \bar{v}$ ist $\}$

Sei also $\Phi := \forall \bar{u}\, \exists \bar{v}\, \varphi$ eine Eingabe für $\Pi_2\text{-SAT}$.

Sei $\bar{u} = u_1 \dots u_\ell$, $\bar{v} = v_1 \dots v_m$.

Für jede Belegung $\bar{y} = y_1 \dots y_\ell \in \{0,1\}^\ell$ für $\bar{u}$ sei $\varphi_{\bar{y}}$ die aussagenlogische Formel, die aus φ entsteht, indem die Variablen $u_1 \dots u_\ell$ durch die Werte $y_1 \dots y_\ell$ ersetzt werden.

Sei $n := |\varphi_{\bar{y}}|$ die Länge der Formel $\varphi_{\bar{y}}$

(Beachte: n hängt nicht von der konkreten Wahl von $\bar{y}$ ab)

Für den SK C'_n aus (*) gilt dann:

(**) Φ ist wahr
$\Leftrightarrow \forall \bar{y} \in \{0,1\}^e$ gilt:
$C'_n(\varphi_{\bar{y}}) = 1$, und der Wert der Gatter $z_1, \dots, z_n$ gibt eine erfüllende Belegung für $\varphi_{\bar{y}}$ an.

Es folgt:

Φ ist wahr
$\Leftrightarrow \exists \gamma \in \{0,1\}^{p(n)} \quad \forall \bar{y} \in \{0,1\}^e$:

(***) γ beschreibt einen SK C' mit n Eingabegattern, einem Ausgabegatter und n speziellen Gattern $z_1, \dots, z_n$, s.d. gilt:
$C'(\varphi_{\bar{y}}) = 1$, und der Wert der Gatter $z_1, \dots, z_n$ gibt eine erfüllende Belegung für $\varphi_{\bar{y}}$ an.

("$\Rightarrow$" gilt wegen (**) und da C'_n durch einen Bitstring der Länge $p(n)$ repräsentiert werden kann;
"$\Leftarrow$" gilt offensichtlicherweise)

Man sieht leicht, dass (***) von einer det. Polyn.zeit-TM überprüft werden kann. Somit ist Π_2-SAT $\in \Sigma_2^P$.

□

Eine TM-basierte Charakterisierung der Klasse P/poly

Definition 6.17 (TM mit "Advice-Strings")

Sei $(\alpha_n)_{n\in\mathbb{N}}$ eine Folge von Bitstrings $\alpha_n \in \{0,1\}^*$.
Sei M eine det. TM und sei $L \subseteq \{0,1\}^*$
Wir sagen: M entscheidet L mit Advice $(\alpha_n)_{n\in\mathbb{N}}$,
falls f.a. $n\in\mathbb{N}$ und f.a. $x\in\{0,1\}^n$ gilt:

$$x \in L \iff M(\langle x, \alpha_n\rangle) = 1$$

(Dh: α_n kann als eine Liste von "Instruktionen" aufgefasst werden, die M bei Eingabe eines Worts x der Länge n befolgt).

Definition 6.18 (Die Klasse $DTIME(T(n))/a(n)$)

Seien $T:\mathbb{N}\to\mathbb{N}$ und $a:\mathbb{N}\to\mathbb{N}$. Die Klasse

$$DTIME(T(n))/a(n)$$

besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, für die gilt:
Es gibt eine Folge $(\alpha_n)_{n\in\mathbb{N}}$ von Bitstrings $\alpha_n \in \{0,1\}^{a(n)}$ und eine det TM M, so dass M die Sprache L mit Advice $(\alpha_n)_{n\in\mathbb{N}}$ entscheidet, und bei Eingabe von $\langle x, \alpha_n\rangle$ nur $O(T(n))$ Schritte macht (für $n\in\mathbb{N}$, $x\in\{0,1\}^n$).

Beispiel 6.19

Jede unäre Sprache $L \subseteq \{1\}^*$ liegt in $DTIME(n)/1$, denn:

Wähle $d_n := \begin{cases} 1 & \text{falls } 1^n \in L \\ 0 & \text{falls } 1^n \notin L \end{cases}$

Sei M eine Linearzeit-TM, die bei Eingabe von $\langle x, d_n \rangle$ für $x \in \{0,1\}^n$ genau dann 1 ausgibt, wenn x nur aus 1en besteht und $d_n = 1$ ist.

Theorem 6.20 (TM-basierte Charakterisierung von P/poly)

$$P/poly = \bigcup_{\substack{c \geq 1, \\ d \geq 1}} DTIME(n^c)/n^d$$

D.h.: P/poly ist die Klasse aller Sprachen, die von det. Polynomialzeit-TMen mit polynomiell langen Advice-Strings entschieden werden.

Beweis:

"$\subseteq$": Sei $L \in P/poly$ und sei $(C_n)_{n \in \mathbb{N}}$ eine SK-Familie polynomieller Größe, die L berechnet. Für jedes $n \in \mathbb{N}$ sei d_n ein Bitstring, der den SK C_n repräsentiert. Sei M eine det. TM, die bei Eingabe von $\langle x, d_n \rangle$ (für $x \in \{0,1\}^n$, $n \in \mathbb{N}$) den SK C_n bei Eingabe x auswertet.

Klar: Das geht in polyn. Zeit. Somit ist $L \in \bigcup_{\substack{c,d \\ \geq 1}} DTIME(n^c)/n^d$

"$\supseteq$": Seien $c, d \geq 1$ und sei $L \in \mathrm{DTIME}(n^c)/n^1$. Sei $(a_n)_{n\in\mathbb{N}}$ mit $a_n \in \{0,1\}^{n^d}$ f.a. $n \in \mathbb{N}$, und sei M eine det. TM, die L mit Advice $(a_n)_{n\in\mathbb{N}}$ entscheidet und f.a. $n \in \mathbb{N}$, $x \in \{0,1\}^n$ bei Eingabe $\langle x, a_n\rangle$ nach $O(n^c)$ Schritten anhält.

Sei $(C_m)_{m\in\mathbb{N}}$ die im Beweis von Theorem 6.13 konstruierte SK-Familie, s.d. f.a. $n \in \mathbb{N}$, $x \in \{0,1\}^n$, $m := |\langle x, a_n\rangle|$ gilt:

$$C_m(\langle x, a_n\rangle) = 1 \iff M(\langle x, a_n\rangle) = 1.$$

C_m lässt sich unter Verwendung von a_n leicht zu einem SK C'_n umbauen, der bei Eingabe von x dasselbe tut wie C_m bei Eingabe $\langle x, a_n\rangle$ (Details: Übung!) Somit ist $(C'_n)_{n\in\mathbb{N}}$ eine SK-Familie der Größe $\mathrm{poly}(m) = \mathrm{poly}(n)$, die L berechnet. D.h.: $L \in P/\mathrm{poly}$. □

6.5 Uniforme Schaltkreis-Familien

Oft betrachtet man nur sog. uniforme SK-Familien $(C_n)_{n\in\mathbb{N}}$, für die es einen effizienten Algorithmus gibt, der bei Eingabe der Zahl n den SK C_n erzeugt. Insbesondere können uniforme SK-Familien daher nur entscheidbare Sprachen berechnen.

Definition 6.21

(a) Eine SK-Familie $(C_n)_{n\in\mathbb{N}}$ heißt P-uniform, falls es eine det. Polynomialzeit-TM gibt, die bei Eingabe eines Worts der Länge n (für $n\in\mathbb{N}$) den SK C_n erzeugt.

(b) Eine SK-Familie $(C_n)_{n\in\mathbb{N}}$ heißt Logspace-uniform, falls die Funktion $f: \{0,1\}^* \to \{0,1\}^*$ mit
$f(x) := C_{|x|}$ (f.a. $x\in\{0,1\}^*$)
write-once logspace-berechenbar ist.

Bemerkung 6.22

(a) Wegen $L \subseteq P$ ist jede logspace-uniforme SK-Familie auch P-uniform.

(b) Wegen Fakt 4.21 ist $(C_n)_{n\in\mathbb{N}}$ genau dann logspace-uniform, wenn jede der folgenden Funktionen auf Platz $O(\log n)$ berechnet werden kann:

- $\text{SIZE}(n) := |C_n|$
- $\text{TYPE}(n,i) :=$ die Markierung (aus $\{x_1,\neg x_1,\dots,x_n,\neg x_n,0,1,\wedge,\vee,?\}$) des i-ten Gatters von C_n
- $\text{EDGE}(n,i,j) :=$ 1 falls es in C_n eine Kante vom i-ten zum j-ten Gatter gibt

(Konvention: Das 0-te Gatter ist das Ausgabegatter)

(c) Für jede P-uniforme SK-Familie $(C_n)_{n\in\mathbb{N}}$ gibt es offensichtlicherweise ein $c\geq 1$ s.d. $|C_n| \leq n^c$ ist (f.a. $n\in\mathbb{N}$)

Definition 6.23

Sei $\mathcal{U} \in \{P, \text{logspace}\}$. Die Klasse

$\mathcal{U}$-uniformes P/poly

besteht aus allen Sprachen $L \subseteq \{0,1\}^*$, die von einer $\mathcal{U}$-uniformen SK-Familie polynomieller Größe berechnet werden

Satz 6.24

logspace-uniformes P/poly $\overset{①}{=}$ P-uniformes P/poly $\overset{②}{=}$ P.

Beweis:

"①, $\subseteq$": Folgt aus Bemerkung 6.22 (a).

"②, $\subseteq$": Sei $(C_n)_{n \in \mathbb{N}}$ eine P-uniforme SK-Familie, die eine Sprache L berechnet

Bei Eingabe von $x \in \{0,1\}^*$ kann eine det. Polynomialzeit-TM wie folgt vorgehen:

(1) Erzeuge den SK C_n für $n := |x|$

(2) Werte C_n mit Eingabe x aus.

Somit ist $L \in P$

"$P \subseteq$ logspace-uniformes P/poly":

Sei $L \in P$.

Aus dem Beweis von Theorem 6.13 erhalten wir für jedes $n \in \mathbb{N}$ einen SK C_n der Größe poly(n), sd. f.a. $x \in \{0,1\}^*$ gilt:

$$x \in L \iff C_n(x) = 1.$$

Gemäß Bemerkung 6.14 ist die SK-Familie $(C_n)_{n\in\mathbb{N}}$ logspace-uniform. □

Bemerkung 6.25 (Nachtrag zu Kapitel 5)

Auf ähnliche Art können wir Satz 5.13 (d): $\boxed{\mathrm{DTIME}(T(n)) \subseteq \mathrm{ASPACE}(\log T(n))}$ (für zeitkonstruierbares $T: \mathbb{N} \to \mathbb{N}$ mit $T(n) \geq n$) beweisen:

Sei $L \in \mathrm{DTIME}(T(n))$ und sei M eine stereotype TM, die L in Zeit $O(T(n)^2)$ entscheidet (vgl. Bemerkung 1.10).

Die Konstruktion aus dem Beweis von Theorem 6.13 liefert eine SK-Familie $(C_n)_{n\in\mathbb{N}}$ der Größe $\mathrm{poly}(T(n))$, s.d. f.a. $n \in \mathbb{N}$ und alle $x \in \{0,1\}^n$ gilt:

$$C_n(x) = 1 \iff M(x) = 1 \iff x \in L.$$

Und gemäß Bemerkung 6.14 kann C_n bei Eingabe von n auf Platz $O(\log T(n))$ berechnet werden (im Sinne von write-once logspace-Berechenbarkeit).

Man kann sich leicht davon überzeugen, dass C_n bei Eingabe von $x \in \{0,1\}^n$ von einer alternierenden TM ausgewertet werden kann, die Platz $O(\log T(n))$ benutzt: $\wedge$- bzw $\vee$-Gatter von C_n werden durch Zustände der TM ausgewertet, die mit "$\forall$" bzw mit "$\exists$" markiert sind. Details: Übung.

Insgesamt erhält man damit: $L \in \mathrm{ASPACE}(\log T(n))$.

□