

Komplexitätstheorie

WiSe 2011/12 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Komplexitätstheorie Gesamtübersicht

- Organisatorisches / Einführung
Motivation / Erinnerung / Fragestellungen
- Diskussion verschiedener Komplexitätsklassen:
Zeitkomplexität
Platzkomplexität
- zugehörige Reduktionsbegriffe
- vollständige Probleme
- Anpassung von Klassenbegriffen und Reduktionen

Die Klasse NL

Satz 1 GAP ist **NL**-vollständig (bzgl. \leq_{\log}).

Da GAP in Polynomzeit lösbar ist, folgt:

Folgerung 2

$$\mathbf{NL} \subseteq \mathbf{P}$$

Da GAP hart ist für **NL**, folgt:

Folgerung 3

$$\mathbf{NL} = \mathbf{L} \iff \text{GAP} \in \mathbf{L}$$

Der Beweis von Satz 1

Wir wissen bereits: $\text{GAP} \in \mathbf{NL}$.

Problem: Gilt $L \leq_{\log} \text{GAP}$ für jedes $L \in \mathbf{NL}$?

Betrachte also $L \in \mathbf{NL}$, $L \subseteq \Sigma^*$. Fixiere nichtdet. 3-Band-TM M mit $L = L_M$ und $S_M(x) \leq \log(\lg(x)) + c$ für $x \in L$.

O.E. beschreibt M nie das Ausgabeband.

\rightsquigarrow Konfigurationen von M auf Eingabe x darstellbar durch:

$$(z, i, w_\ell, a, w_r),$$

wobei $z \in Z$, $0 \leq i \leq \lg(x) + 1$ (Eingabekopfposition),

$w_\ell a w_r \in \Gamma^*$ ist "aktueller" Arbeitsbandinhalt.

M logspace-beschränkt \rightsquigarrow

Wir können $\lg(w_\ell a w_r) = \lg(w_\ell) + 1 + \lg(w_r) \leq c \log(\lg(x)) + c$ annehmen.

Der Beweis von Satz 1 (Forts.)

Wir konstruieren einen Graphen $G(M, x)$ zu (M, x) mit obigen Konfigurationen als Knotenmenge $V(M, x)$ (plus ausgezeichnetem Zielknoten ζ). Kanten aus $E(M, x)$ reflektieren den Einschnitt-Übergang; zusätzlich führe Kanten von Endkonfigurationen (aus $V(M, x) \cap \mathcal{K}_e$) zu ζ ein.

Offenbar: $x \in L = L_M$ g.d.w. In $G(M, x)$ ist ζ von $I_M(x)$ aus erreichbar.

D.h.: $f : x \mapsto$ Darstellung von $G(M, x)$ ist eine many-one Reduktion.

Frage: Gilt $f \in \text{FSPACE}(\log)$?

Erzeugung der Adjazenzmatrix durch nur $\mathcal{O}(\log(\lg(x)))$ Platz benötigende Zähler (einen für die Zeilen i , einen für die Spalten j); in logarithmischem Platz kann dann weiter geprüft werden, ob $K_i \vdash_M K_j$ gilt (gültiger Konfigurationsübergang).

Entsprechend werden Einsen und Nullen (und Klammern) ausgegeben.

Die Behandlung der Endkonfigurationen ist trivial.

Beachte, dass die "Parameter" von M als Konstanten eingehen (zu L existiert. . .)

Problem ZUSAMMENHANG:

- Gegeben gerichteter Graph $G = (V, E)$ mit $V = \{1, \dots, n\}$
- Frage: Ist G stark zusammenhängend?

Erinnerung: *starker Zusammenhang* meint Existenz von (gerichteten) Pfaden zwischen je zwei Knoten.

Lemma 4 ZUSAMMENHANG *ist NL-vollständig (bzgl. \leq_{\log}).*

Teil 1 ZUSAMMENHANG liegt in NL:

Idee: Benutze bekannten Algorithmus für GAP , genauer:

Betrachte Variante, bei der zwei Knoten x, y auf dem Arbeitsband notiert sind und die Adjazenzmatrix auf dem Eingabeband steht.

Der Algorithmus für $ZUSAMMENHANG$ betreibt zwei Zähler für x und y auf dem Arbeitsband und fragt den GAP -Algorithmus für jede Kombination, ob ein Pfad von x nach y existiert.

Genau dann, wenn jeder Aufruf von GAP JA zurückliefert, ist der Eingabegraph stark zusammenhängend.

Beachte: Die zusätzlichen Zähler x und y benötigen nur logarithmischen Platz in der Größe der Adjazenzmatrix.

Teil 2 ZUSAMMENHANG ist hart für NL:

Wir zeigen: $\text{GAP} \leq_{\log} \text{Zusammenhang}$.

Seien $G = (V, E)$ und $x, y \in V$ gegeben.

Konstruiere $G' = (V', E')$ durch $V' = V$ und

$$E' = E \cup \{y\} \times V \cup V \times \{x\}$$

Beh.: G' stark zusammenhängend \iff In G gibt es Pfad von x nach y .

Außerdem: Die Transformation ist deterministisch mit log. Platz ausführbar.

\Leftarrow : Dann gibt es für beliebige $u, v \in V'$ in G' Pfad u, x, \dots, y, v .

\Rightarrow : Insbesondere: Es gibt Pfad von x nach y in G' .

Betrachte so einen Pfad P kürzester Länge \rightsquigarrow

Weder x noch y kommen als "Zwischenpunkte" auf P vor.

\rightsquigarrow Die in G' neu eingeführten Kanten aus y heraus und auch nicht die Kanten nach x hinein kommen nicht in P vor.

\rightsquigarrow P ist ein Pfad von x nach y in G .

Spiele auf Graphen über Markierung verschiedener Knoten
(analog zu Mühle-Spiel, Solitaire,...)

unterschiedlichen Markierungsregeln

⇒ Vollständige Probleme verschiedener Klassen!

Einfachster Regelsatz für ein *Pebble-Spiel*:

- Ein Knoten ohne Vorgänger kann jederzeit markiert werden.
- Ein Knoten mit Vorgängern kann markiert werden, wenn mindestens einer seiner Vorgänger markiert ist.

Anmerkung: x 'Vorgänger' von y , wenn (x, y) Kante ist.

Problem EINFACHES PEBBLE:

- Gegeben sei ein gerichteter Graph $G = (V, E)$ mit $V = \{1, \dots, n\}$
- Frage: Kann mit dem obigen Pebble-Spiel in G ein Knoten ohne Nachfolger markiert werden?

Lemma 5 EINFACHES PEBBLE *ist NL-vollständig (bzgl. \leq_{\log}).*

Hinweis: Vollständigkeit von EINFACHES PEBBLE als Übungsaufgabe

Problem co-GAP :

Tupel (G, x, y) , für die in G kein Pfad von x nach y existiert.

Mit $\text{GAP} \in \mathbf{P}$ und Lemma aus voriger Vorlesung über Komplementabschluss von \mathbf{P} folgt:

$$\text{co-GAP} \in \mathbf{P}$$

$\text{GAP} \in \mathbf{NL}$: leicht einsehbar (s.o.)

$\text{co-GAP} \in \mathbf{NL}$: ?!

20 Jahre offenes Problem, bis 1987 (N. Immermann / R. Szelepcsényi)

Beide gelten als Erfinder des *induktiven Zählens*.

Sie haben ihre Ergebnisse unabhängig voneinander gefunden, für Szelepcsényi war das seine Diplomarbeit (!)

co-GAP \in NL Hinführung / Vorüberlegung

Sei $G = (V, E)$ ein gerichteter Graph, $x, y \in V$ gegeben.

Sei $V = \{1, \dots, n\}$. Setze

$$V_m^x = \{z \in V \mid \exists z_0, \dots, z_j, j \leq m : z_0 = x, z_j = z, \forall 1 \leq i \leq j : (z_{i-1}, z_i) \in E\}.$$

In Worten: Menge der in höchstens m Schritten von x aus erreichbaren Knoten.

Klar: $y \in V_{n-1}^x$ gdw. y ist von x aus erreichbar.

Im Folgenden: Es werden drei nichtdeterministische Hilfsprogramme definiert, die jeweils auf (G, x, y) zugreifen und jeweils höchstens $\mathcal{O}(\log(n))$ Platz benötigen. Der Nichtdeterminismus wird durch **Mitzählen** der Rateschritte gezäumt.

co-GAP \in NL 1. Hilfsprogramm

gap1 erhält als Parameter z und m und liefert *nichtdeterministischen Test*, ob $z \in V_m^x$ gilt.

Das bedeutet: gap1 liefert 1 oder 0; 1 wird genau dann geliefert, wenn ein Pfad von x nach z der Länge $\leq m$ **durch Raten gefunden wurde**.

Das Ergebnis 0 bedeutet also nicht unbedingt, dass kein solcher Pfad existiert.

Es werden drei (logspace) Hilfsvariablen p, q, j benötigt.

```
p ← x;  
FOR j ← 1 TO m DO  
  IF p ≠ z THEN  
    rate q ∈ V;  
    IF (p, q) ∈ E THEN p ← q  
  ENDFOR  
gap1 ← (p = z).
```

Hinweis: Nichtdet. TM werden “normalerweise” nicht zum Berechnen von Funktionen definiert, da “Ausgabe” unklar.

co-GAP \in NL 2. Hilfsprogramm

test erhält als Parameter z , m und k .

Dabei **Annahme**: $k = |V_m^x|$.

test liefert:

1, falls $z \in V_m^x$ durch Raten bemerkt wurde.

0, falls $z \notin V_m^x$ durch Raten bemerkt wurde;

test divergiert, falls "Fehlraten" festgestellt werden konnte.

Zusätzlich zu den Eingabe werden

zwei Hilfsvariablen a, \hat{z} benötigt:

a : # gefundener Knoten aus V_m^x .

\hat{z} : Hilfszählknoten

$a \leftarrow 0$; $test \leftarrow 0$;

FOR $\hat{z} \leftarrow 1$ TO n DO

 IF $gap1(\hat{z}, m)$ THEN

$a \leftarrow a + 1$

 IF $z = \hat{z}$ THEN $test \leftarrow 1$;

IF $a \neq k$ THEN divergiere

Beobachte: (1) Ist die Annahme richtig, so kann test durch **Mitzählen selbst** bemerken, ob je falsch geraten wurde.

(2) Ist die Annahme falsch, liefert die Prozedur *irgendetwas* in den letzten beiden Fällen; insbesondere die korrekte Rückgabe von 0 fußt auf der Annahme.

Problem: Bestimmung von $|V_m^x|$; **Leicht** für $m = 0 \rightsquigarrow$ **induktiv ?!**

co-GAP \in NL 3. Hilfsprogramm

Jetzt kommt die Induktion ins Spiel !

count erhält als Parameter m und k .

Dabei **Annahme**: $k = |V_{m-1}^x|$.

count liefert:

$|V_m^x|$, falls Annahme richtig und immer richtig geraten wurde.

irgendeine Zahl, falls Annahme falsch und immer richtig geraten wurde

count divergiert, falls "Fehlraten" festgestellt werden konnte.

Zusätzlich zu den Eingabe werden drei Hilfsvariablen a, \hat{z} , gefunden benötigt:

a : # gefundener Knoten aus V_m^x .

\hat{z} : Hilfszählknoten.

gefunden: Boolesche Variable

$a \leftarrow 0$;

FOR $\hat{z} \leftarrow 1$ TO n DO

IF test($\hat{z}, m-1, k$) THEN

$a \leftarrow a + 1$

ELSE

gefunden $\leftarrow 0$;

FOR $z \leftarrow 1$ TO n DO

IF test($z, m-1, k$) \wedge (z, \hat{z}) $\in E$ THEN

gefunden $\leftarrow 1$;

IF gefunden THEN $a \leftarrow a + 1$

count $\leftarrow a$

$\text{co-GAP} \in \mathbf{NL}$ Das Hauptprogramm `co-gap`

Korrektheit folgt durch Induktion !

`co-gap` erhält als Parameter G sowie x und y .

`co-gap` verhält sich wie folgt:

konvergent, falls y nicht von x aus erreicht werden kann und immer richtig geraten wurde.

divergent, falls fehlgeraten wurde oder falls y von x aus erreicht werden kann.

Zusätzlich zu den Eingabe werden zwei Hilfsvariablen m, k benötigt:
 m : mögliche Länge eines Pfades von x nach y
 k : Hilfszähler; Invariante: $k = |V_m^x|$.

```
m ← 0; k ← 1;
FOR m ← 1 TO n - 1 DO
  k ← count(m, k)
  // Stets richtig geraten ? ~> Invariante.
  // Jemals falsch geraten ? ~> Divergenz !
IF test(y, n - 1, k) THEN
  Divergiere
```

Beobachte: (1) Alle Hilfsvariablen benötigen logarithmischen Platz.

(2) Die Rekursion ist "konstant tief".

Lemma 6

$$\text{co-GAP} \in \mathbf{NL}$$

Folgerung 7

$$\mathbf{NL} = \text{co-NL}$$

Anmerkung: Als Folgerung (ohne Beweis) für viele Komplexitätsschranken s :

$$\text{NSPACE}(s) = \text{co-NSPACE}(s)$$

Folgerung für formale Sprachen:

Satz 8 *Kontextsensitive Sprachen ($= \text{NSPACE}(n)$) sind unter Komplementbildung abgeschlossen.*

Zum Korollar:

Es gilt: $A \leq B \iff$

$$(\exists f : \forall x : x \in A \iff f(x) \in B) \iff (\exists f : \forall x : x \in \bar{A} \iff f(x) \in \bar{B}) \iff \text{co-}A \leq \text{co-}B$$

für geeignete Reduktionsbegriffe \leq und passende Reduktionen f .

Aus der Def. von co-NL folgt daher:

$$A \in \text{co-NL} \iff \text{co-}A \leq_{\log} \text{GAP} \iff A \leq_{\log} \text{co-GAP}$$

Mit $\text{co-GAP} \in \text{NL}$ folgt $A \in \text{NL}$.

Da $A \in \text{co-NL}$ beliebig: $\sim \text{NL} \supseteq \text{co-NL}$. Daher ferner:

$B \in \text{NL} \iff \text{co-}B \in \text{co-NL} \subseteq \text{NL} \sim B \in \text{co-NL}$. Also: $\text{NL} \subseteq \text{co-NL}$.