Die Erreichbarkeits-Methode

Theorem: Sei $f(n)$ eine ordentliche Komplexitätsfunktion. Dann gelten:

(a) **SPACE**$(f(n)) \subseteq$ **NSPACE**$(f(n))$ und **TIME**$(f(n)) \subseteq$ **NTIME**$(f(n))$

(b) **NTIME**$(f(n)) \subseteq$ **SPACE**$(f(n))$

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

Beweis:

(a) Jede TM ist auch eine NTM. $\surd$

(b) Simulation aller nichtdeterministischen Auswahlmöglichkeiten in Platz $f(n)$ (mit Platz-Recycling). $\surd$

(c) Mittels Erreichbarkeits-Methode:

▷

Zu zeigen: (c) **NSPACE**$(f(n)) \subseteq$ **TIME**$(k^{\log n + f(n)})$:

- Sei M eine k\$ NTM mit I/O, die L in Platz $f(n)$ entscheidet. Wir entwickeln nun eine deterministische Methode, um die nichtdeterministische Berechnung von M bei Eingabe von x in $c^{\log n + f(n)}$ Schritten zu simulieren, wobei $n = |x|$ und c nur von M abhängt.
- Sei $G(M,x)$ der Konfigurationsgraph von M: Es gibt eine Kante von Knoten (dh der Konfiguration) C_1 zu Knoten C_2 gdw $C_1 \stackrel{M}{\to} C_2$.
- $x \in L$ gdw es zumindest einen Pfad in $G(M,x)$ von $C_0 = (s, \triangleright, x, \triangleright, \varepsilon, \ldots, \triangleright, \varepsilon)$ zu einer Konfiguration der Form $C = (\text{„}ja\text{“}, \ldots)$ gibt.

▷

- Eine Konfiguration $(q, w_1, u_1, \ldots, w_k, u_k)$ ist eine Momentaufnahme einer Berechnung.
- Bei I/O TM, von denen man ja nur eine „ja“/„nein“ Antwort auf die Frage „$x \in L$?“ erwartet, spielt der write-only output-\$ keine Rolle.
- Beim read-only input-\$ interessiert uns auch nur die Position $0 \leq i \leq n$ des Cursors.
- Für alle $k-2$ anderen \$s ist die Länge höchstens $f(n)$.
- Jede Konfiguration kann daher als $(q, i, w_2, u_2, \ldots, w_{k-1}, u_{k-1})$ dargestellt werden.
- Wie viele Konfigurationen kann M insgesamt haben?
- Antwort: Höchstens $|K|(n+1)|\Sigma|^{(2k-2)f(n)} \leq c_1^{\log n + f(n)}$, wobei c_1 nur von M abhängt.

▷

Die Frage „$x \in L$?“ kann daher entschieden werden, indem man das Erreichbarkeitsproblem auf einem Graphen mit höchstens $c_1^{\log n + f(n)}$ Knoten löst. Für das Erreichbarkeitsproblem gibt es quadratische Algorithmen ($\to$ shortest path $\in$ **P**), dh die Simulation braucht insgesamt höchstens $c_2 c_1^{2(\log n + f(n))} = c^{\log n + f(n)}$ Schritte mit $c = c_2 c_1^2$. $\surd$

Bemerkung: Der Graph $G(M,x)$ muss nicht explizit repräsentiert werden. Stattdessen kann der Algorithmus jedesmal, wenn er wissen will, ob es eine Kante von C nach C' gibt, dies implizit aus den Konfigurationen C und C', der Eingabe x und der Maschinenbeschreibung von M neu berechnen.

Korollar: **L** $\subseteq$ **NL** $\subseteq$ **P** $\subseteq$ **NP** $\subseteq$ **PSPACE** $\subseteq$ **EXP**

Beweis:

- **L** $=$ **SPACE**$(\log n)$ $\subseteq$ **NSPACE**$(\log n)$ $=$ **NL** wegen (a).
- **NL** $=$ **NSPACE**$(\log n)$ $\subseteq$ **TIME**$(k^{\log n+\log n})$ $=$ **TIME**$(n^{2\log k})$ $\subseteq$ **P** wegen (c).
- **TIME**(n^k) $\subseteq$ **NTIME**(n^k) $\Rightarrow$ **P** $\subseteq$ **NP** wegen (a).
- **NTIME**(n^k) $\subseteq$ **SPACE**(n^k) $\Rightarrow$ **NP** $\subseteq$ **PSPACE** wegen (b).
- **SPACE**(n^k) $\subseteq$ **NSPACE**(n^k) wegen (a) und **NSPACE**(n^k) $\subseteq$ **TIME**$(c^{\log n+n^k})$ $\subseteq$ **TIME**$(2^{(n^{k+c'})})$ wegen (c), dh **SPACE**(n^k) $\subseteq$ **TIME**$(2^{(n^{k'})})$ $\Rightarrow$ **PSPACE** $\subseteq$ **EXP**. $\surd$

Aus dem Platz-Hierarchie Theorem folgt:

Korollar: **L** $\subset$ **PSPACE** .

Beweis: **L** $=$ **SPACE**$(\log n)$ $\subset$ **SPACE**$(\log n \log(\log n))$ $\subseteq$ **SPACE**(n^2) $\subseteq$ **PSPACE** . $\surd$

- Vermutlich sind alle Inklusionen zwischen den Komplexitätsklassen des Korollars auf Seite 5 strikt, wir wissen jedoch nur, dass
- zwischen **L** und **PSPACE** zumindest eine Inklusion strikt ist (allerdings nicht, welche), und dass
- zwischen **P** und **EXP** zumindest eine Inklusion strikt ist (allerdings wieder nicht, welche).

NSPACE

- Wie effizient lässt sich nichtdeterministischer Platz durch deterministischen Platz simulieren?
- Vom Theorem auf Seite 1 wissen wir, dass

  NSPACE$(f(n))$ $\subseteq$ **SPACE**$(c^{\log n+f(n)})$

- Frage: Geht es auch besser als mit exponentiellem Mehraufwand?

  Antwort: Ja, und zwar quadratisch, mittels folgendem

Theorem: (Savitch 1970)

Erreichbarkeit $\in$ **SPACE**$((\log n)^2)$

▷

Beweis: Gegeben ein Graph G und Knoten x, y sowie eine Zahl $i \geq 0$. Sei PFAD(x,y,i) $\Leftrightarrow$ „es gibt einen Pfad von x nach y, der höchstens 2^i Kanten lang ist.“

Bemerkung: Wenn der Graph n Knoten enthält, dann ist jeder Pfad höchstens n lang und wir können das Erreichbarkeitsproblem in G lösen, wenn wir PFAD$(x,y,\lceil \log n \rceil)$ berechnen können. Dies ist wie folgt möglich:

```
Funktion pfad(x,y,i)
if i = 0 then
   if x = y or (x,y) ∈ E(G) then return „ja“
else for jeden Knoten z do      /* Mittel-Suche */
        if pfad(x,z,i−1) and pfad(z,y,i−1)
        then return „ja“;
return „nein“
```

▷

• Wenn $i = 0$, dann pfad$(x,y,i) =$ PFAD(x,y,i). Für $i > 0$ ist pfad$(x,y,i) =$ „ja“ gdw es einen Knoten z gibt, sodass sowohl pfad$(x,z,i-1)$ als auch pfad$(z,y,i-1)$ gelten. Wegen der Induktions-Hypothese gilt dann, dass es Pfade von x nach z und von z nach y gibt, die höchstens 2^{i-1} Kanten lang sind. Daher gibt es einen Pfad von x nach y, der höchstens $2 \times 2^{i-1} = 2^i$ Kanten lang ist.

• Den $O((\log n)^2)$ Platzbedarf erreichen wir, indem wir die Rekursion über einen Stack realisieren, der für jeden aktiven Aufruf ein Triplett (x,y,i) speichert: Für jeden Knoten z schreibe $(x,z,i-1)$ auf den Stack und rufe pfad$(x,z,i-1)$ auf. Falls dies „ja“ liefert, wird $(x,z,i-1)$ mit $(z,y,i-1)$ überschrieben, sonst mit $(x,z',i-1)$.

• Da $i \leq \lceil \log n \rceil$, können höchstens $\lceil \log n \rceil$ viele aktive Tripletts gleichzeitig am Stack stehen, wovon jedes höchstens $3 \log n$ Platz benötigt $\rightarrow O((\log n)^2)$. $\surd$

Korollar: **NSPACE**$(f(n)) \subseteq$ **SPACE**$((f(n))^2)$ für beliebige ordentliche Komplexitätsfunktion $f(n) \geq \log n$.

Beweis:

- Um eine $f(n)$ Platz-beschränkte NTM M mit Eingabe x zu simulieren, lassen wir den pfad Algorithmus auf dem Konfigurations-Graphen $G(M,x)$ laufen.
- Der Graph ist nur implizit repräsentiert, dh wir brauchen ihn nicht zu speichern. Es genügt, die Eingabe x und die Beschreibung von M lesen zu können, wenn $i = 0$.
- Der Konfigurations-Graph hat höchstens $c^{f(n)}$ Knoten, daher brauchen wir insgesamt höchstens $O((\log(c^{f(n)}))^2) = O((f(n))^2)$ viel Platz. $\surd$

Korollar: **PSPACE** $=$ **NPSPACE**; offen: **L** $\stackrel{?}{=}$ **NL**

Alle NSPACE Komplexitätsklassen sind unter Komplement abgeschlossen

- Schlüsselkonzept: Die Anzahl der von Knoten x erreichbaren Knoten kann in **NSPACE**$(\log n)$ berechnet werden.
- Auch das Komplement davon (die Anzahl der von Knoten x nicht erreichbaren Knoten) kann in **NSPACE**$(\log n)$ berechnet werden.
- Wie berechnet eine NTM M eine Funktion von \$s zu \$s? $\rightarrow$ Mindestens eine Berechnung liefert das korrekte Resultat, die anderen divergieren.

Theorem: (Immerman-Szelepscényi 1988 $\rightarrow$ Gödel-Preis 1995) Gegeben ein Graph G und ein Knoten x. Die Anzahl der von x erreichbaren Knoten in G kann in **NSPACE**$(\log n) =$ **NL** berechnet werden.

Beweis: Wir berechnen iterativ die Zahlen $|S(1)|, |S(2)|, \ldots, |S(n-1)|$, wobei $S(k)$ die Menge der Knoten in G ist, die in Pfaden der Länge höchstens k von x aus erreichbar sind.

$|S(n-1)|$ ist dann die Gesamtanzahl der von x erreichbaren Knoten in G. $\triangleright$