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? $\to$ Mindestens eine Berechnung liefert das korrekte Resultat, die anderen divergieren.

Theorem: (Immerman-Szelepscényi 1988 $\to$ 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. ▷

Algorithmus CountReachableNodes(G,n,x)

G ist ein gerichteter Graph, implizit durch die Funktion $G(i,j)$ gegeben, die **wahr** ist gdw es eine Kante von Knoten i zu j gibt oder $i = j$.
n ist die # der Knoten in G.
x ist der Startknoten.

```
C_prev := 1;
for k := 1 to n−1 do
/* C_prev: #Knoten/von x in k−1 Schritten erreichbar sind */
  C_curr := 0;
  for y := 1 to n do
    /* Ist y von x in ≤ k Schritten erreichbar? */
    if isElement(x,y,G,k,C_prev) then C_curr := C_curr + 1;
  C_prev := C_curr;
Schreibe C_prev auf den Output$.
```

▷

Non-det. Algorithmus isElement(x,y,G,k,C_{prev})

$\boxed{x \xrightarrow{k-1} v \longrightarrow y}$

```
sc := 0; reply := false;
for v := 1 to n do
  goodPath := true; u_curr := x;
  for p := 1 to k−1 do /* Rate einen k−1 langen Pfad */
    u_prev := u_curr; guess u_curr;
    if not G(u_prev, u_curr) then
      goodPath := false;
  /* ... und checke, ob der Pfad „gut“ ist und in v endet */
  if goodPath and u_curr = v then sc := sc + 1;
    /* Wenn y außerdem von v erreichbar ist ... */
    if G(v,y) then reply := true;
if sc < C_prev then “Non-det. abort!”
else return reply.
```

▷

- Der Algorithmus „läuft" in **NL** und braucht dabei nur konstant viele Variable, jede mit logarithmisch viel Platzbedarf.
- Wenn C_{prev} bereits korrekt berechnet wurde (Anfangbedingung), dann wird C_{curr} erhöht $\Leftrightarrow$ $y \in S(k)$ $\Leftrightarrow$ die Berechnung bricht nicht non-deterministisch ab, der „sanity-counter" $sc = C_{prev}$, und $G(v,y)$.
- Für jeden Knoten v wird ein Pfad von x zu v der Länge $k-1$ geraten. Wenn es insgesamt weniger als C_{prev} solche v's gibt, erkennen wir, dass nicht gut geraten wurde, und die Berechnung wird verworfen. Umgekehrt: wenn $sc = C_{prev}$, dann haben wir alle in Frage kommenden v's gefunden und wissen daher auch sicher, ob einer davon zu y führt und daher, dass y von x in höchstens k Schritten erreichbar ist. $\surd$

Korollar: Wenn $f(n) \geq \log n$ eine ordentliche Komplexitätsfunktion ist, dann

$$\textbf{NSPACE}(f(n)) = \textbf{coNSPACE}(f(n)).$$

Beweis: Angenommen $L \in$ **NSPACE**$(f(n))$ wird durch eine $f(n)$-Platz-beschränkte NTM M entschieden. Wir geben eine NTM $\overline{M}$ die $\overline{L}$ enscheidet an.

- Bei Eingabe von w lässt $\overline{M}$ den vorherigen Algorithmus auf dem Konfigurationsgraphen $G(M,w)$ laufen (implizit repräsentiert).
- Wenn $\overline{M}$ dabei eine akzeptierende Konfiguration in $S(k)$ findet, so hält er und verwirft w. Wenn umgekehrt bis $k = 2^{O(f(|w|))} - 1$ alle möglichen Pfadlängen in $G(M,w)$ überprüft wurden, ohne dass eine akzeptierende Konfiguration erreichbar gewesen wäre, dann akzeptiert $\overline{M}$ die Eingabe w (dh $w \notin L$).

$\triangleright$

`Algorithmus` Unreachability(G,n,x,a)

G,n,x**:** wie bei CountReachableNodes.
a**:** akzeptierenden Konfigurationen von M.

$C_{prev} := 1$;
for $k := 1$ **to** $n-1$ **do**
/* C_{prev}: #Knoten/von x in $k-1$ Schritten erreichbar sind */
  $C_{curr} := 0$;
  for $y := 1$ **to** n **do**
    /* Ist y von x in $\leq k$ Schritten erreichbar? */
    if isElement(x,y,G,k,C_{prev}) **then**
      $C_{curr} := C_{curr} + 1$;
      $\boxed{\textbf{if } y \in a \textbf{ then } \text{REJECT;}}$
  $C_{prev} := C_{curr}$;
$\boxed{\text{ACCEPT.}}$

$\triangleright$

- $G(M,w)$ hat höchstens $2^{O(f(|w|))}$ Knoten, daher braucht $\overline{M}$ höchstens $\log 2^{O(f(|w|))} = O(f(|w|))$ Platz. $\surd$
- Folgte für $f(n) \geq \text{poly}(n)$ schon aus Savitchs Theorem, damit aber auch klar für:

Korollar: **NL** $=$ **coNL**; weiterhin offen: **L** $\overset{?}{\neq}$ **NL**

Bemerkung:

- Erreichbarkeit (und Nichterreichbarkeit) sind **NL**-vollständig.
- 2SAT (und 2UNSAT) sind **NL**-vollständig.
- Simple Logic Programming (Regeln der Form $p \to q$ und Fakten) ist **NL**-vollständig.