Große Übung 5

1. Komplexitätsanalyse

$L = \{ w\#u\#v \mid w \in u ⧢ v \}$

Der Shuffle $u ⧢ v$ ist eine Menge

$\varepsilon ⧢ v = v ⧢ \varepsilon = \{v\}$

$(u.a) ⧢ (v.b) = (u ⧢ (v.b))a \cup ((u.a) ⧢ v)b$

Bsp: $\boxed{011010}\#001\#110$

Idee 1:

```
input w#u#v
for i ∈ {0,1}^|w|          // 2^|w| ∈ 2^Θ(n) Iterationen → 2^Θ(n)·Θ(n²+n²) = 2^Θ(n)·2^Θ(n) = 2^Θ(n)
  if alle Buchstaben w_k mit 0 = i_k matchen mit u
    (σ_{i=0}(zip(w,i)) = u), then                        // Θ(n²)
      if σ_{i=1}(zip(w,i)) = v, then return 1            // Θ(n²) (bzw. Θ(n) mit 2. Band)
  end if
end for
return 0
```

$\sigma_{i=0} : (\{0,1\} \times \{0,1\})^* \to (\{0,1\})^*$ (Paare (w, i))

$\sigma_{i=0}(0,0) = 0 \qquad (1,0) = 1$

$(0,1) = \varepsilon \qquad (1,1) = \varepsilon$

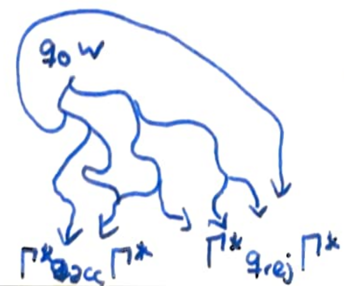
Zeitkomplexität für 1 Band: $\Gamma = \{0, 1, \#, 0', 1', ␣\}$

$\to L \in DTIME(2^{\Theta(n)}) \cap DTIME_2(2^{\Theta(n)})$

Auch für 2 Bänder

Für NTIME darf der Algorithmus i erraten.

Nichtdeterministische Entscheider M $w \in L(M)$:



$w \notin L(M)$

input $w\#u\#v$ $\quad \forall$ input, finite

guess $i \in \{0,1\}^{|w|}$ $\quad // \Theta(n)$

if $\sigma_{i=0}(zip(w,i)) = u$, **then** $\quad // \Theta(n^2)$ bzw. $\Theta(n)$

  if $\sigma_{i=1}(zip(w,i)) = v$, **then** $\quad // \Theta(n^2)$ bzw. $\Theta(n)$

    accept

  end if

end if

reject

$\hookrightarrow L \in NTIME(\Theta(n^2)) \cap NTIME_2(\Theta(n))$

Algo 3: Idee: Speichere den Fortschritt in w, u, v mit Binärzählern.

Rate i_k buchstabenweise

input $w\#u\#v$ $\quad k_v \leftarrow 0$, $k_u \leftarrow 0$

for $k \leq |w|$

  guess $i_k \in \{0,1\}$

  if $i_k = 0$ **then** **if** $w_k \neq u_{k_u}$ **then** **reject** **else** $k_u{+}{+}$

  else **if** $w_k \neq v_{k_v}$ **then** **reject** **else** $k_v{+}{+}$

end for

accept

$k \leq |w| \in \Theta(n)$

$k_v \leq |v| \in \Theta(n)$

$k_u \leq |u| \in \Theta(n)$

$\hookrightarrow \Theta(3 \cdot \log n)$ Bits

$\rightarrow NSPACE(\Theta(\log n))$

Für DSPACE, Betrachte Idee/Algo 1. Dieser nutzt keine zusätzlichen Bandzellen

$\rightarrow L \in DSPACE(\Theta(n))$

Auch für NSPACE haben alle Ausführungen von Algo 2 höchstens $\Theta(n)$ Platzbedarf

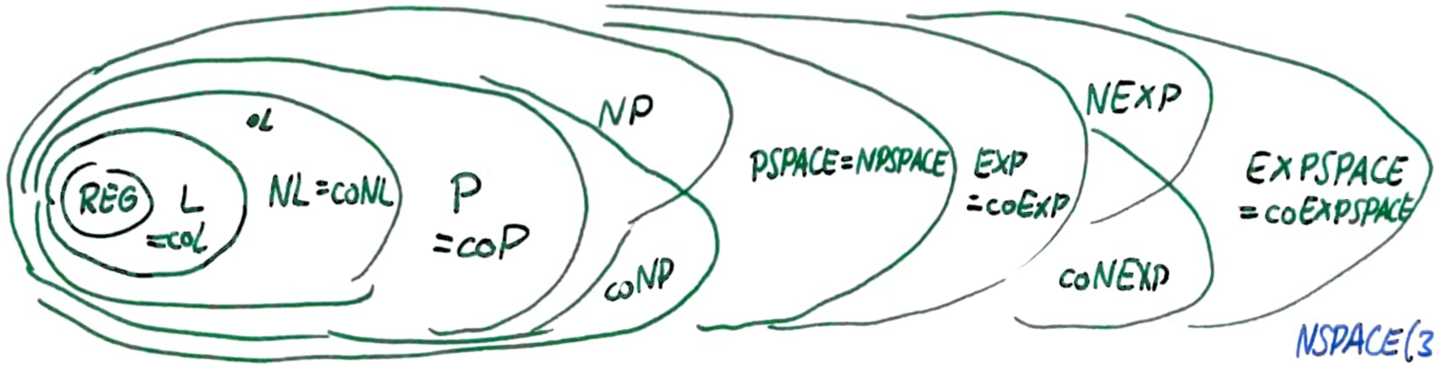
$\rightarrow L \in NSPACE(\Theta(n))$

$L \in DTIME_1(2^{\Theta(n)}) \cap DTIME_2(2^{\Theta(n)}) \cap NTIME_1(\Theta(n^2)) \cap NTIME_2(\Theta(n)) \cap DSPACE(\Theta(n)) \cap NSPACE(\Theta(n))$

$\subseteq EXP \qquad \cap NP \qquad \cap PSPACE \qquad \cap NPSPACE$

$\cap NL$

$\subseteq P$



$\Rightarrow$ Es gibt auch einen Algorithmus mit Poly. Zeitkompl.

$NSPACE(3 \log n) \subseteq DTIME(2^{3 \log n})$
$= DTIME(n^3)$

$NL = \{ L(M) \mid M \text{ Entscheider mit } NSPACE_M(n) \le c \cdot \log n \ \ (\exists c \in \mathbb{N}, \forall n \in \mathbb{N}) \}$

ACYCPATH

Eingabe: gerichteter Graph $\langle V, \to \rangle$, $s, t \in V$
Zusicherung: $\langle V, \to \rangle$ enthält keine Kreise
Frage: Gibt es einen Pfad von s nach t ?

PATH $\in$ NL

```
input ⟨V,→⟩, s∈V, t∈V
counter ← 0, v ← s
while counter ≤ |V|   { if v=t, accept
   guess w∈V
   if v ↛ w, reject
   v ← w
   counter ++
end
reject
```

counter $\le |V| \in \Theta(n)$
$|v| \le \log |\langle V, \to \rangle|$
$\to NSPACE(\Theta(\log n))$

Korrektheit:

Falls $\langle \langle V, \to \rangle, s, t \rangle \in PATH$, dann gibt es einen s-t-Pfad mit Länge $\le |V|$
→ Der Algo. kann diese Knotenfolge erraten. Die Berechnung, die das tut, erkennt in jeder Iteration $v \to w$, und dass der letzte Knoten $= t$ ist. Diese Berechnung akzeptiert.
→ Algo. akzeptiert.

Falls Algo $\langle \langle V, \to \rangle, s, t \rangle$ akzeptiert, dann tat dies eine Berechnung. Diese erriet einen ~~Weg~~ s-t-Weg der Länge $\le |V|$. Falls der Weg einen ~~Wa~~Knoten doppelt besucht hat, gibt es einen kürzeren Weg, also letztendlich einen s-t-Pfad
→ $\in PATH$

Nächste Woche

ACYCPATH ist NL-vollständig bzgl. LogSpace m.o.R.

ACYCPATH $\le_m^{\log}$ PATH, (mit $f = id$)

also ist PATH NL-vollständig.

wegen NL = coNL, $\overline{PATH}$ ist NL-vollständig. 2SAT auch.