Der Satz von Savitch

Sommerakademie Rot an der Rot — AG 1
"Wieviel Platz brauchen Algorithmen wirklich?"

Peter Faymonville

Reactive Systems Group
Universität des Saarlandes

9. August 2010

Übersicht

Wiederholung

Definition: REACH für Graph G und Knoten s,t

- **Frage:** Gibt es einen Pfad in G von s nach t?
- **Antwort:** Ja/Nein

Das Erreichbarkeitsproblem (REACH) für gerichtete Graphen...

- ...liegt in NL.
- ...ist NL-vollständig.

Walter J. Savitch

- 1969: PhD in Mathematik, UC Berkeley (Betreuer: Stephen Cook)
- seit 1969: Professor (jetzt Emeritus), UC San Diego
- Komplexitätstheorie, Computerlinguistik
- Lehrbücher: Ada, Java, C++

Ein quadratisch-logarithmischer Algorithmus für REACH

Idee

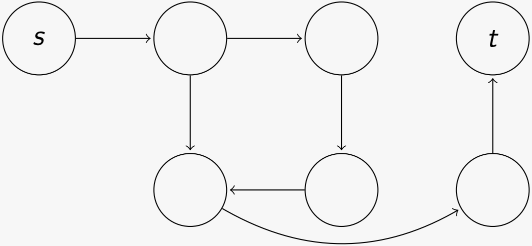
Jeder Pfad hat einen äquidistanten Mittelpunkt.

Definition

$\textsc{reachMax}\ (s, t, k) \Leftrightarrow \exists$ Pfad von s nach t mit Maximallänge 2^k

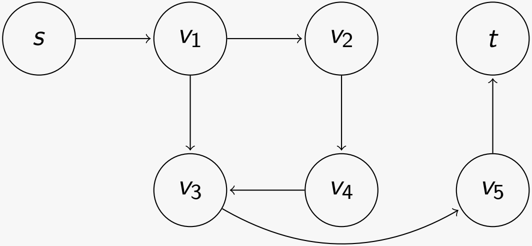
function $\textsc{reachMax}(s,\ t,\ k)$: Boolean
    if $k < 0$ **then return** false
    else if $(s, t) \in E$ **then return** true
    else
        for $v \in V - \{s, t\}$ **do**
            if reachMax(s,v,$k-1$) **then**
                if reachMax(v,t,$k-1$) **then**
                    return true
    return false

Algorithmus für REACH – Visualisierung



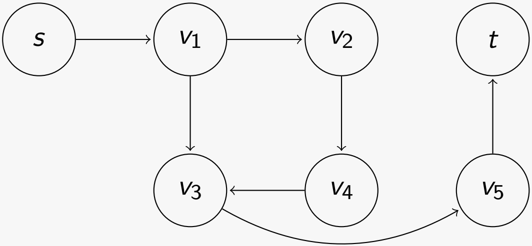
REACHMAX $(s, t, 2)$?

Algorithmus für REACH – Visualisierung



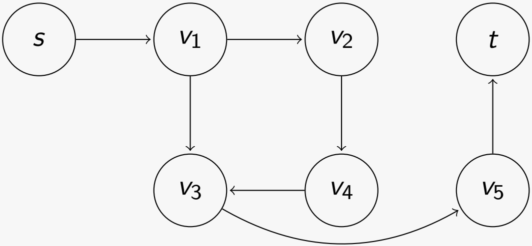
$\textsc{reachMax}\ (s, t, 2)?$
$\rightarrow (s, v_1, 1)?, (v_1, t, 1)?$

Algorithmus für REACH – Visualisierung



reachMax $(s, t, 2)$?
$\rightarrow (s, v_1, 1)?, (v_1, t, 1)?$
$\rightarrow (s, v_1, 1)!, (v_1, t, 1)?$

Algorithmus für REACH – Visualisierung



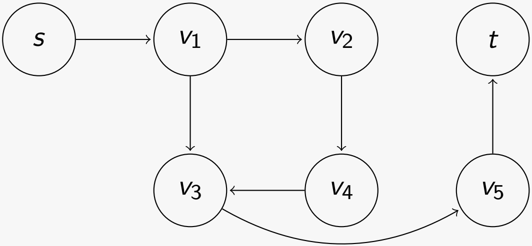
$\textsc{reach}\mathrm{Max}\ (s, t, 2)?$

$\longrightarrow (s, v_1, 1)?, (v_1, t, 1)?$

$\longrightarrow (s, v_1, 1)!, (v_1, t, 1)?$

$\longrightarrow (s, v_1, 1)!, (v_1, v_3, 0)?(v_3, t, 0)?$

Algorithmus für REACH – Visualisierung



$\textsc{reach}\mathrm{Max}\ (s, t, 2)?$

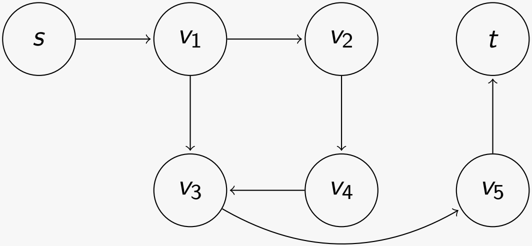
$\longrightarrow (s, v_1, 1)?, (v_1, t, 1)?$

$\longrightarrow (s, v_1, 1)!, (v_1, t, 1)?$

$\longrightarrow (s, v_1, 1)!, (v_1, v_3, 0)?(v_3, t, 0)?$

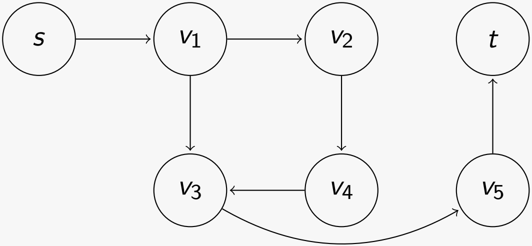
$\longrightarrow (s, v_1, 1)!, (v_1, v_3, 0)!(v_3, t, 0)?$

Algorithmus für REACH – Visualisierung



REACHMAX $(s, t, 2)$?
$\rightarrow (s, v_3, 1)?, (v_3, t, 1)?$

Algorithmus für REACH – Visualisierung

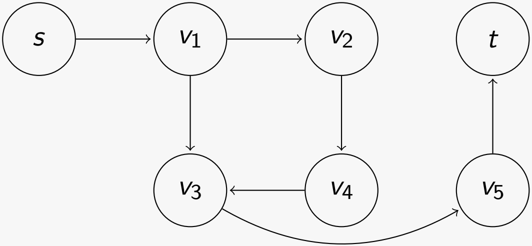


REACHMAX $(s, t, 2)$?

$\rightarrow (s, v_3, 1)?, (v_3, t, 1)?$

$\rightarrow (s, v_1, 0)?, (v_1, v_3, 0)?, (v_3, t, 1)?$

Algorithmus für REACH – Visualisierung



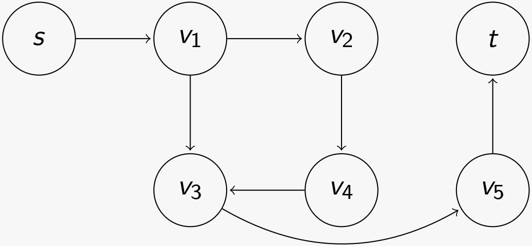
$\textsc{reachMax}\ (s, t, 2)?$

$\rightarrow (s, v_3, 1)?, (v_3, t, 1)?$

$\rightarrow (s, v_1, 0)?, (v_1, v_3, 0)?, (v_3, t, 1)?$

$\rightarrow (s, v_1, 0)!, (v_1, v_3, 0)!, (v_3, t, 1)?$

Algorithmus für REACH – Visualisierung



REACHMAX $(s, t, 2)$?

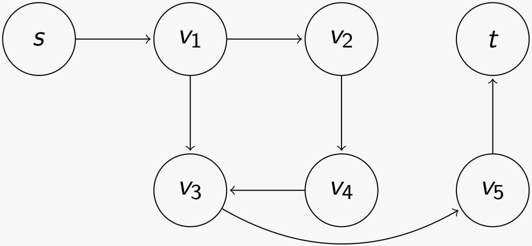
$\longrightarrow (s, v_3, 1)?, (v_3, t, 1)?$

$\longrightarrow (s, v_1, 0)?, (v_1, v_3, 0)?, (v_3, t, 1)?$

$\longrightarrow (s, v_1, 0)!, (v_1, v_3, 0)!, (v_3, t, 1)?$

$\longrightarrow (s, v_1, 0)!, (v_1, v_3, 0)!, (v_3, v_5, 0)?, (v_5, t, 0)?$

Algorithmus für REACH – Visualisierung



$\textsc{reachMax}\ (s, t, 2)?$

$\longrightarrow (s, v_3, 1)?, (v_3, t, 1)?$

$\longrightarrow (s, v_1, 0)?, (v_1, v_3, 0)?, (v_3, t, 1)?$

$\longrightarrow (s, v_1, 0)!, (v_1, v_3, 0)!, (v_3, t, 1)?$

$\longrightarrow (s, v_1, 0)!, (v_1, v_3, 0)!, (v_3, v_5, 0)?, (v_5, t, 0)?$

$\longrightarrow (s, v_1, 0)!, (v_1, v_3, 0)!, (v_3, v_5, 0)!, (v_5, t, 0)!$

Algorithmus für REACH – Korrektheit

```
function REACHMAX(s, t, k): Boolean
    if  k < 0  then return false
    else if  (s, t) ∈ E then return true
    else
        for v ∈ V − {s, t}  do
            if reachMax(s,v,k − 1) then
                if reachMax(v,t,k − 1) then
                    return true
    return false
```

- Basisfall: Wenn $k = 0$, teste direkte Erreichbarkeit über eine Kante
- Rekursion: Wenn $k > 0$, iteriere über alle möglichen Mittelpunkte.

Beweis

Satz von Savitch

$$\mathrm{REACH} \in \mathsf{SPACE}(O(\log^2 n))$$

Beweis

Satz von Savitch

$$\mathrm{REACH} \in \mathsf{SPACE}(O(\log^2 n))$$

Realisierung als Mehrband-DTM

- Repräsentation von G liegt auf dem Eingabeband
- Arbeitsband 1: Aktuell betrachtetes v (foreach)
- Arbeitsband 2: Rekursionsstack

Beweis

Satz von Savitch

$$\mathrm{REACH} \in \mathsf{SPACE}(O(\log^2 n))$$

Realisierung als Mehrband-DTM

- Repräsentation von G liegt auf dem Eingabeband
- Arbeitsband 1: Aktuell betrachtetes v (foreach)
- Arbeitsband 2: Rekursionsstack

Analyse des Speicherplatzes

- Rekursionstiefe: $\log n$ Aufrufe (Halbierung der Distanz)
- Größe des Aufrufs: $3 \times \log n$ (Kodierung s,t,k)

Konsequenzen

$$\mathbf{NL} \subseteq \mathrm{SPACE}(O(\log^2 n))$$

$$\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{L}^2$$

Verhältnis von NSPACE und SPACE

Idee

Jede NDTM M kann durch eine quadratische Simulation determinisiert werden.

1. Ausrollen des Konfigurationsgraphen von M
2. Erreichbarkeit der Endzustände mittels REACH testen

Nichtdeterministische Turingmaschinen

NDTM M akzeptiert Sprache L

- Eingabeband, Ausgabeband
- s Arbeitsbänder
- Zustandsraum Q
- c Elemente des Bandalphabets
- n Länge der Eingabe w

Konfigurationsgraph für w

- Graph ist azyklisch
- $|Q|$ Zustände
- $c^{s+2\cdot f(n)}$ Bandkonfigurationen, wenn $L \in \text{NSPACE}(f(n))$
- $n \cdot f(n)^{s+1}$ Bandkopfpositionen

Anzahl der Konfigurationen

$$|Q| \cdot c^{s+2\cdot f(n)} \cdot n \cdot f(n)^{s+1}$$

Beweis

$$\text{NSPACE}(f(n)) \subseteq \text{SPACE}(f^2(n))$$

für eine ordentliche[1] Komplexitätsfunktion $f(n) \geq O(\log\ n)$.

1. Gegeben eine NDTM M, die eine Sprache $L \in \text{NSPACE}(f(n))$ akzeptiert.
2. Modifiziere Savitch's Algorithmus:
3. Beim Test auf $(s, t) \in E$, benutze Transitionsrelation von M.
4. Der Konfigurationsgraph hat $k^{\log\ n + f(n)}$ Knoten, für k konstant.
5. Daher reicht $O(f^2(n))$ Platz, um die Erreichbarkeit von Endzuständen zu entscheiden.

[1] monoton steigend, berechenbar von einer TM in $O(|n| + f(n))$ Zeit, $O(f(n))$ Platz

Konsequenzen

$$\text{NPSPACE} = \text{PSPACE}$$

Zusammenfassung

- Es gibt einen quadratisch-logarithmischen Algorithmus für **gerichtete** Erreichbarkeit.
- Nichtdeterministischer Platz und deterministischer Platz fallen für polynomiellen Platz zusammen.

Vielen Dank für ihre Aufmerksamkeit!