8. Ein Algorithmus für die transitive Hülle in Digraphen mit linearer erwarteter Laufzeit

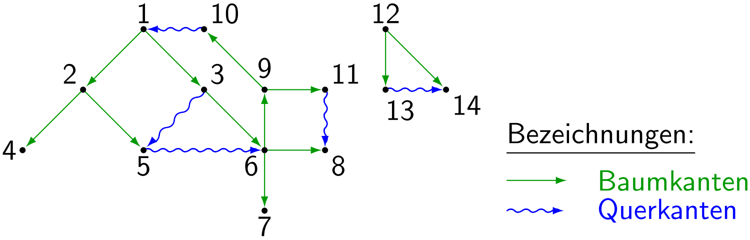
Wir nehmen an, dass die Wahrscheinlichkeit eines Digraphen mit n Knoten und m Kanten eine Funktion (nur) von n und m ist. Daraus folgt (wir lassen der Einfachheit halber Schleifen (= Kanten $v \to v$) zu), dass jede Kante (u, v) mit Wahrscheinlichkeit $\frac{m}{n^2}$ vorhanden ist, falls wir Digraphen mit n Knoten und m Kanten betrachten.

Erinnerung: Breitensuche (BFS): Schlange, queue, FIFO:

$$\longrightarrow \boxed{\text{FIFO}} \longrightarrow$$

Durchlaufe Graphen, indem wir, solange möglich, den vordersten Knoten v aus der Queue nehmen, ihn behandeln und die Liste seiner noch nicht behandelten Nachbarn in die Queue hinten einfügen.

Beispiel 122



algorithm exp-lin-transitive-closure

0. Konstruiere die linear geordneten Adjazenzlisten L_i^r, $i = 1, \ldots, n$ des Graphen G^r (entsteht aus G durch Umkehrung aller Kanten).
   Beispiel:

$$
\begin{array}{lcccc|lcccc}
L_1 & = & 4 & 1 & 2 & L_1^r & = & 1 & 2 & 4 \\
L_2 & = & 1 & 4 & 3 & L_2^r & = & 1 & 3 & 4 \\
L_3 & = & 3 & 2 & & L_3^r & = & 2 & 3 & \\
L_4 & = & 1 & 4 & 2 & L_4^r & = & 1 & 2 & 4
\end{array}
$$

   Ersetze ebenfalls alle L_i durch L_i^{rr} ($\rightarrow$ sortierte Adjazenzlisten)

1. Berechne für jeden Knoten i in BFS-Art eine Liste S_i von von i aus erreichbaren Knoten, so dass (i) oder (ii) gilt:
   (i) $|S_i| < \left\lfloor \frac{n}{2} \right\rfloor + 1$ und S_i enthält alle von i aus erreichbaren Knoten
   (ii) $|S_i| = \left\lfloor \frac{n}{2} \right\rfloor + 1$
2. Entsprechende Listen P_i von Vorgängern von i
3. **for** $i := 1$ **to** n **do**
   $L_i^* := S_i \cup \{j; i \in P_j\} \cup \{j;\ |S_i| = |P_j| = \left\lfloor \frac{n}{2} \right\rfloor + 1\}$
   od
   Bilde $(L_i^*)^{rr}$ für $i = 1, \ldots, n$

Korrektheit: $j \in L_i^*$ gdw (i, j) in transitiver Hülle.

Satz 123

Sei die Wahrscheinlichkeit eines Graphen (lediglich) eine Funktion der Anzahl n seiner Knoten und der Anzahl m seiner Kanten. Dann ist der Erwartungswert für die Laufzeit des obigen Algorithmus $\mathcal{O}(n + m^*)$. Dabei ist m^* der Erwartungswert für die Anzahl der Kanten in der transitiven Hülle.

Beweis:

Das Durchlaufen des Graphen (von v_1 aus für die Konstruktion von S_1) in BFS-Manier liefert eine Folge $(a_t)_{t\geq 1}$ von Knoten. Sei $L_{\sigma(\nu)}$ die ν-te Adjazenzliste, die in der BFS erkundet wird,

$$L_{\sigma(\nu)} = (a_t;\ h(\nu) < t \leq h(\nu+1))$$

Sei $\Delta L_{\sigma(\nu)}$ die Menge der a_t in $L_{\sigma(\nu)}$, und sei (für $i = 1$)

$$S_i(t) := \{a_1, a_2, \dots, a_t\}.$$

Wie bereits gezeigt, ist

$$\Pr[j \in L_i] = \frac{m}{n^2}, \quad \forall i, j$$

Beweis (Forts.):

Alle n-Tupel von geordneten Listen $L_1, L_2, \ldots, L_n$ mit

$$m = \sum_{i=1}^{n} |L_i|$$

sind aufgrund der Voraussetzungen über die Wahrscheinlichkeitsverteilung gleich wahrscheinlich. Also:

Beobachtung 1: Die Mengen $\Delta L_{\sigma(\nu)}$, $\nu = 1, 2, \ldots$ sind (paarweise) unabhängig.

Beobachtung 2: Die $\Delta L_{\sigma(\nu)}$ sind gleichverteilt, d.h. für alle $A \subseteq \{1, \ldots, n\}$ gilt:

$$\Pr[\Delta L_{\sigma(\nu)} = A] = \left(\frac{m}{n^2}\right)^{|A|}\left(1 - \frac{m}{n^2}\right)^{n-|A|}$$

Beweis (Forts.):

Lemma 124

Sei $A \subseteq \{1, \ldots, n\}$ *mit* $|A| = k$. *Sei* $(\bar{a}_t)_{t \geq 1}$ *eine Folge von unabhängigen gleichverteilten Zufallsvariablen mit Wertemenge* $\{1, \ldots, n\}$. *Dann gilt:*

$$\min\{t; \bar{a}_t \notin A\} \text{ hat Erwartungswert } 1 + \frac{k}{n-k}.$$

Beweis:

[des Lemmas] $\Pr[\bar{a}_1, \ldots, \bar{a}_r \in A \text{ und } \bar{a}_{r+1} \notin A] = \left(\frac{k}{n}\right)^r \cdot (1 - \frac{k}{n})$.
Also ist der Erwartungswert für $\min\{t; \bar{a}_t \notin A\}$:

$$\begin{aligned}
1 + \sum_{r=0}^{\infty} r \left(\frac{k}{n}\right)^r \left(1 - \frac{k}{n}\right) &= 1 + \frac{k}{n} \left(1 - \frac{k}{n}\right) \sum_{r \geq 1} r \left(\frac{k}{n}\right)^{r-1} \\
&= 1 + \frac{k}{n} \left(1 - \frac{k}{n}\right) \frac{1}{\left(1 - \frac{k}{n}\right)^2} \\
&= 1 + \frac{\frac{k}{n}}{1 - \frac{k}{n}} = 1 + \frac{k}{n-k} \,.
\end{aligned}$$

□

Anmerkung:

$$\sum_{r=0}^{\infty} r z^{r-1} = \frac{d}{dz} \frac{1}{1-z} = \frac{1}{(1-z)^2}$$

Lemma 125

Sei $(\bar{a}_t)_{t\geq 1}$ *wie oben,* $k \leq \frac{n}{2}$*. Dann hat* $\min\{t; |\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_t| > k\}$ *Erwartungswert* $\leq \frac{3}{2}(k+1)$*.*

Beweis:

[des Lemmas] Wegen der Additivität des Erwartungswertes gilt: Der gesuchte Erwartungswert ist

$$
\begin{aligned}
&\leq \sum_{\nu=0}^{k}\left(1+\frac{\nu}{n-\nu}\right)\\
&\leq k+1+\sum_{\nu=1}^{k}\frac{\nu}{\frac{n}{2}}\\
&\leq k+1+\frac{k(k+1)}{n} \leq \frac{3}{2}(k+1)\,.
\end{aligned}
$$

$\square$

Beweis (Forts.):

Wenn wir jedes $L_{\sigma(\nu)}$ in sich zufällig permutieren, erhalten wir eine Folge $(\bar{a}'_t)_{t\geq 1}$ von Zufallsvariablen, so dass

$$|\{\bar{a}_1, \ldots, \bar{a}_t\}| = |\{\bar{a}'_1, \ldots, \bar{a}'_t\}| \text{ für } t = h(\nu), \quad \forall \nu$$

Da die $\bar{a}'_t$ innerhalb eines jeden Intervalls $h(\nu) < t \leq h(\nu + 1)$ paarweise verschieden sind, gilt für sie die vorhergehende Abschätzung erst recht. Wir betrachten nun aus Symmetriegründen o.B.d.A. lediglich den Knoten $i = 1$. Dann sind $(|S_1(t)|)_{t\geq 1}$ und $(|S_1(t)'|)_{t\geq 1}$ gleichverteilte Folgen von Zufallsvariablen, denn dies gilt zunächst für alle Zeitpunkte der Form $t = h(y)$; da aber für diese Zeitpunkte $S_1(t) = S'_1(t)$ ist und $h(\ )$ zufällig verteilt ist, ist auch die Verteilung der beiden Folgen zwischen diesen Intervallgrenzen identisch.

Beweis (Forts.):

Sei s der Erwartungswert und $|S_i|$ die tatsächliche Größe von S_i nach Phase 1. Dann

$$s = \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor + 1} k \cdot \Pr[|S_i| = k]\,.$$

Die erwartete Anzahl der Schritte (und damit die Kosten) in der BFS sei q. Dann gilt:

$$q \leq \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor + 1} \Pr[|S_i| = k] \frac{3}{2} k = \frac{3}{2} s.$$

Also ist der Aufwand des Algorithmus für die Phasen 1 und 2 im Erwartungswert $\leq 3sn$. Da $ns \leq m^*$ und die Kosten der anderen Phasen offensichtlich durch $\mathcal{O}(n + m + m^*)$ beschränkt sind, folgt die Behauptung. $\square$

C.P. Schnorr:
An algorithm for transitive closure with linear expected time
SIAM J. Comput. **7**(2), pp. 127–133 (1978)

9. Matrixmultiplikation à la Strassen

Betrachte das Produkt zweier 2×2-Matrizen:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}; \quad c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j}$$

Man beachte, dass diese und die folgenden Formeln nicht die Kommutativität der Multiplikation voraussetzen. Sie gelten also auch, falls die a_{ij}, b_{ij}, c_{ij} eigentlich $n \times n$-Matrizen A_{ij}, B_{ij}, C_{ij} sind (jeweils $i, j \in \{1, 2\}$).

Bilde:

$$
\begin{aligned}
m_1 &:= (a_{12} - a_{22})(b_{21} + b_{22}) \\
m_2 &:= (a_{11} + a_{22})(b_{11} + b_{22}) \\
m_3 &:= (a_{11} - a_{21})(b_{11} + b_{12}) \\
m_4 &:= (a_{11} + a_{12})b_{22} \\
m_5 &:= a_{11}(b_{12} - b_{22}) \\
m_6 &:= a_{22}(b_{21} - b_{11}) \\
m_7 &:= (a_{21} + a_{22})b_{11}
\end{aligned}
$$

Dann gilt:

$$
\begin{aligned}
c_{11} &:= m_1 + m_2 - m_4 + m_6 \\
c_{12} &:= m_4 + m_5 \\
c_{21} &:= m_6 + m_7 \\
c_{22} &:= m_2 - m_3 + m_5 - m_7
\end{aligned}
$$

Sei $n = 2^k$ und $M(n)$ die Anzahl arithmetischer Operationen (Mult, Add, Sub), die Strassen's Algorithmus bei $n \times n$ Matrizen benötigt:

$$\begin{aligned} M(1) &= 1 \\ M(n) &= 7M\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 \end{aligned}$$

Expansion dieser Rekursionsgleichung $\Rightarrow$

$$\begin{aligned} M(n) &= 7^{k+1} - 6n^2 = 7 \cdot 2^{k \log_2 7} - 6n^2 \\ &= 7n^{\log_2 7} - 6n^2 \\ &< 7n^{2{,}80736} - 6n^2 \\ &= \mathcal{O}(n^{2{,}81})\,. \end{aligned}$$

Der Exponent ω für die Matrixmultiplikation ist also $< 2,81$.

proc $\textsc{MatMult}(A, B, n) =$

co A und B sind $n \times n$ Matrizen **oc**

if $n < 16$ **then** berechne $A \cdot B$ mit klassischer Methode

elif n gerade **then**

spalte A und B in je 4 $\frac{n}{2} \times \frac{n}{2}$ Matrizen auf und wende Strassen's Formeln an. Führe die Multiplikationen rekursiv mit $\textsc{MatMult}$ aus.

else

spalte A und B in eine $(n-1) \times (n-1)$ Matrix (A_{11}, B_{11}) und drei verbleibende Blöcke auf. Wende $\textsc{MatMult}$ rekursiv auf A_{11}, B_{11} an und berechne die anderen Produkte mit der klassischen Methode.

fi

Satz 126

Der MATMULT*-Algorithmus hat folgende Eigenschaften:*

1. *Für $n < 16$ ist die Anzahl der arithmetischen Operationen genauso hoch wie bei der klassischen Methode.*
2. *Für $n \geq 16$ ist die Anzahl der arithmetischen Operationen echt kleiner als bei der klassischen Methode.*
3. MATMULT *braucht nie mehr als $4,91n^{\log_2 7}$ arithmetische Operationen.*

Beweis:

1. $\surd$

Beweis (Forts.):

2. Sei n gerade und seien A, B zwei $n \times n$ Matrizen. Wir wenden Strassen's Formeln an, führen aber die 7 Multiplikationen der $\frac{n}{2} \times \frac{n}{2}$ Matrizen mit der klassischen Methode aus.
   Gesamtzahl arithmetischer Operationen:
   $$7\left(2\left(\frac{n}{2}\right)^3 - \left(\frac{n}{2}\right)^2\right) + 18\left(\frac{n}{2}\right)^2 = \frac{7}{4}n^3 + \frac{11}{4}n^2$$
   Dies ist besser als die klassische Methode, falls:
   $$\frac{7}{4}n^3 + \frac{11}{4}n^2 < 2n^3 - n^2 \Leftrightarrow \frac{15}{4}n^2 < \frac{1}{4}n^3 \Leftrightarrow 15 < n \Leftrightarrow n \geq 16$$
   Sei n ungerade. Multiplizieren wir $A \cdot B$ mit der klassischen Methode, so auch $A_{11} \cdot B_{11}$. Also brauchen wir durch Anwendung von Strassen's Formeln auf die $(n-1) \times (n-1)$ Matrizen ($n-1$ gerade) weniger Operationen, wenn $n - 1 \geq 16$ ist.

Beweis (Forts.):

Sei $M'(n)$ die Anzahl arithmetischer Operationen in MATMULT.
Dann ist:

$$M'(n) = \begin{cases} 2n^3 - n^2 & \text{falls } n < 16 \\ 7M'\left(\frac{n}{2}\right) + \frac{18}{4}n^2 & \text{falls } n \geq 16 \text{ gerade} \\ 7M'\left(\frac{n-1}{2}\right) + \frac{42}{4}n^2 - 17n + \frac{15}{2} & \text{falls } n \geq 16 \text{ ungerade} \end{cases}$$

Wir definieren für $x \in \mathbb{R}^+$:

$$\bar{M}(x) = \begin{cases} 2x^3 - x^2 & \text{falls } x < 32 \\ 7\bar{M}\left(\frac{x}{2}\right) + \frac{42}{4}x^2 & \text{falls } x \geq 32 \end{cases}$$

Dann gilt:

$$\bar{M}(n) \geq M'(n) \text{ für alle } n \in \mathbb{N}\,.$$

Beweis (Forts.):

Es ist

$$\bar{M}(x) = \sum_{i=0}^{k-1} \left[7^i \cdot \frac{42}{4} \left(\frac{x}{2^i}\right)^2\right] + 7^k \left(2 \cdot \left(\frac{x}{2^k}\right)^3 - \left(\frac{x}{2^k}\right)^2\right)$$

für $x \geq 32$, wobei $k := \min\left\{l \,\middle|\, \frac{x}{2^l} < 32\right\}$.

Mit $k = \lfloor \log_2 x \rfloor - 4 =: \log_2 x - t$ erhalten wir $t \in [4, 5[$ und

$$\bar{M}(x) \leq 7^{\log_2 x} \left(14 \left(\frac{4}{7}\right)^t + 2 \left(\frac{8}{7}\right)^t\right) \leq 4,91 \cdot 7^{\log_2 x} \text{ für } x \geq 32.$$

Für $x < 32$ direkt nachrechnen. □

Volker Strassen:
Gaussian elimination is not optimal
Numer. Math. **13**, pp. 354–356 (1969)

Victor Ya. Pan:
New fast algorithms for matrix operations
SIAM J. Comput. **9**(2), pp. 321–342 (1980)

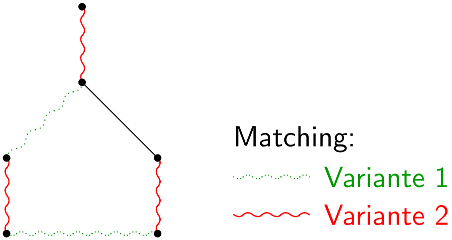
Don Coppersmith, Shmuel Winograd:
Matrix multiplication via arithmetic progressions
J. Symbolic Computation **9**(3), pp. 251–280 (1990)

Kapitel VI Matchings in Graphen

1. Grundlagen

Definition 127

Sei $G = (V, E)$ ein ungerichteter, schlichter Graph. Ein Matching M in G ist eine Teilmenge von E, so dass keine zwei Kanten aus M einen Endpunkt gemeinsam haben.



Definition 128

1. Ein Matching M in $G = (V, E)$ heißt perfekt (oder vollkommen), falls
$$|M| = \frac{|V|}{2}$$
2. Ein Matching M heißt „Matching maximaler Kardinalität“ (engl. maximum matching) in G, falls es in G kein Matching M' mit $|M'| > |M|$ gibt (durchgehend geringelt/rot im Beispiel)
3. Ein Matching M heißt maximal in G, falls es bezüglich „$\subseteq$“ maximal ist (engl. maximal matching) (gepunktet geringelt/grün im Beispiel)

Definition 129

Sei $G = (V, E)$ ein Graph. Wenn V in zwei nichtleere Teilmengen V_1 und V_2 partitioniert werden kann $(V_1 \cup V_2 = V, V_1 \cap V_2 = \emptyset)$, so dass $E \subseteq V_1 \times V_2$ ist, dann heißt G bipartit $(G = (V_1, V_2, E))$.