Kapitel 2: Die AGM-Schranke und andere obere Schranken für die Größe von Anfrageergebnissen

Der Begriff "AGM-Schranke" ist benannt nach den Autoren Atserias, Grohe und Marx des Artikels "Size bounds and query plans for relational joins" In Proc. FoCS 2008, pp. 739-748.

2.1 Warm-Up: Die Δ-Anfrage

Betrachte die Anfrage Q_Δ aus Beispiel 1.2

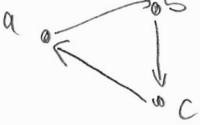
$$Q_\Delta(A,B,C) \leftarrow E(A,B), E(B,C), E(C,A).$$

Sei $\sigma := \sigma_E = \{E\}$. Jede σ-DB $\mathcal{D}$ entspricht einem gerichteten Graphen $G^{\mathcal{D}} = (V^{\mathcal{D}}, E^{\mathcal{D}})$ mit $V^{\mathcal{D}} := adom(\mathcal{D})$.

Klar: Für jede σ-DB $\mathcal{D}$ ist

$$Q_\Delta(\mathcal{D}) = \{ (a,b,c) : (a,b) \in E^{\mathcal{D}} \text{ und } (b,c) \in E^{\mathcal{D}} \text{ und } (c,a) \in E^{\mathcal{D}} \}$$

Skizze:



D.h: $Q_\Delta(\mathcal{D})$ besteht aus allen gerichteten Dreiecken in $G^{\mathcal{D}}$.

Ziel: Gib eine obere Schranke für die Anzahl $|Q_\Delta(\mathcal{D})|$ von Tupeln in $Q_\Delta(\mathcal{D})$ an und finde einen möglichst effizienten Algorithmus, der $Q_\Delta(\mathcal{D})$ berechnet.

Notation: $n^D := |adom(D)|$

$N^D := |E^D|$

Offensichtliche obere Schranken für $|Q_\Delta(D)|$:

1) $|Q_\Delta(D)| \leq (n^D)^3$

2) $|Q_\Delta(D)| \leq (N^D)^2$

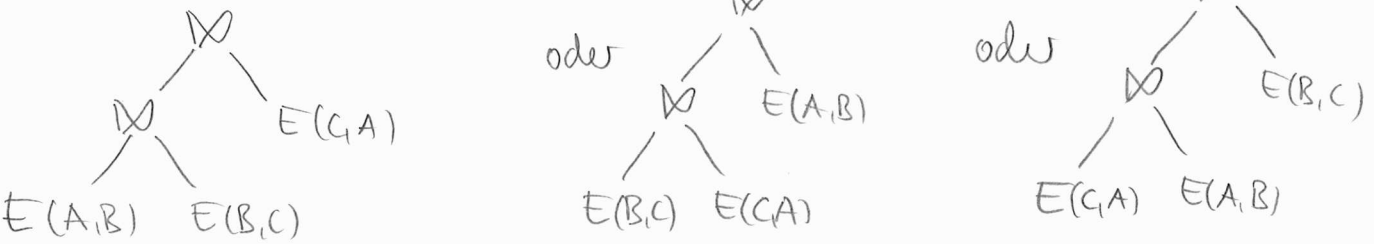
Frage: Geht das besser?

Antwort, die der nächste Satz (Satz Δ) gibt:

Ja: $|Q_\Delta(D)| \leq 2 \cdot (N^D)^{1,5}$ — und es gibt auch einen Algorithmus, der $Q_\Delta(D)$ mit Laufzeit $\approx (N^D)^{1,5}$ löst.

Bevor wir uns den Satz (inkl. Beweis) anschauen, diskutieren wir aber erstmal, wie ein herkömmliches Datenbanksystem die Anfrage Q_Δ auswertet (siehe Beispiel 1.1 für eine Formulierung der Anfrage in SQL).

Um die Anfrage Q_Δ auszuwerten wird ein Datenbanksystem i.d.R. QEPs (query evaluation plans) der folgenden Art betrachten



Bei jedem dieser QEPs wird als Zwischenergebnis $Q_2(D)$ berechnet, für

$$Q_2(X,Y,Z) \leftarrow E(X,Y), E(Y,Z)$$

und dann wird

$$Q_\Delta(X,Y,Z) \leftarrow Q_2(X,Y,Z), E(Z,X)$$

auf D ausgewertet.

Beispiel 2.1

Für jede Zahl $m \geq 1$ gibt es eine 0-DB D_m mit

$N^{D_m} = \Theta(m)$, $|Q_\Delta(D_m)| = \Theta(m)$, aber

$|Q_2(D_m)| = \Omega(m^2)$ für $Q_2(X,Y,Z) \leftarrow E(X,Y), E(Y,Z)$.

Die "üblichen" QEPs (query evaluation plans), die in Datenbanksystemen zur Auswertung der Anfrage Q_Δ erzeugt werden, benötigen zum Auswerten von Q_Δ auf D_m daher Zeit $\Omega((N^{D_m})^2)$, während der Algorithmus aus dem folgenden Satz Δ nur Zeit $O((N^{D_m})^{1,5})$ benötigt.

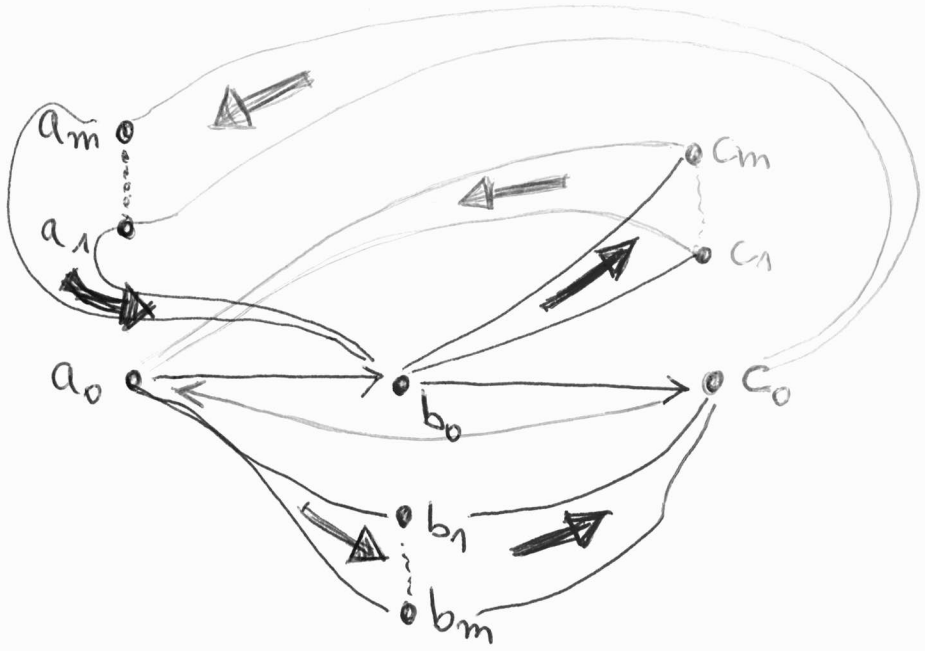
Konstruktion von D_m:

Seien $a_0, a_1, \ldots, a_m$, $b_0, b_1, \ldots, b_m$, $c_0, c_1, \ldots, c_m$ $3(m+1)$ verschiedene Elemente aus dom. Sei

$$
\begin{aligned}
E^{D_m} := \; & \{ (a_0, b_i) : i \in [0,m] \} \cup \\
& \{ (a_i, b_0) : i \in [1,m] \} \cup \\
& \{ (b_0, c_i) : i \in [0,m] \} \cup \\
& \{ (b_i, c_0) : i \in [1,m] \} \cup \\
& \{ (c_0, a_i) : i \in [0,m] \} \cup \\
& \{ (c_i, a_0) : i \in [1,m] \}
\end{aligned}
$$

Klar: $N^{D_m} := |E^{D_m}| = 3 \cdot (m+1+m) = 6m+3$

Skizze: D_m



Man kann sich leicht davon überzeugen, dass gilt:

$$Q_\Delta(D_m) = X \cup \pi_{2,3,1}(X) \cup \pi_{3,1,2}(X) \text{ für}$$

$$X = \{(a_0, b_0, c_i) : i \in [0,m]\} \cup \{(a_0, b_i, c_0) : i \in [1,m]\} \cup \{(a_i, b_0, c_0) : i \in [1,m]\}.$$

Es gilt: $|X| = m+1+2m = 3m+1$ und $|Q_\Delta(D_m)| = 3|X| = 9m+3$

Außerdem ist $(a_i, b_0, c_j) \in Q_2(D_m)$, f.a. $i,j \in [0,m]$,

also $|Q_2(D_m)| > m^2$

Satz Δ: Für jede σ-DB D gilt:

$$|Q_\Delta(D)| \leq 2 \cdot (N^D)^{3/2}$$

und es gibt einen Algorithmus, der bei Eingabe von D die Menge $Q_\Delta(D)$ in Zeit $O((N^D)^{3/2} \cdot t)$ berechnet, wobei t die Zeit ist, die wir benötigen, um bei Eingabe von (v,w) zu testen, ob $(v,w) \in E^D$ ist.

Beweis: Sei D eine beliebige σ-DB.

Für jedes $v \in adom(D)$ sei

$$aus\text{-}Grad^D(v) := |\{w : (v,w) \in E^D\}|$$
$$= |E^D(v,*)|$$

für $E^D(v,*) := \{w : (v,w) \in E^D\}|$.

Ein Knoten $v \in adom(D)$ heißt

- **schwer**, wenn $aus\text{-}Grad^D(v) \geq \sqrt{N^D}$ ist
- **leicht**, wenn v nicht schwer ist.

Ein Tupel $t = (a,b,c) \in Q_\Delta(D)$ heißt
schwer, wenn a schwer ist, und es heißt
leicht, wenn a leicht ist.

Für $X \in \{\text{leicht}, \text{schwer}\}$ sei

$$Q_{\Delta,X}(\mathbb{D}) := \{ t \in Q_\Delta(\mathbb{D}) : t \text{ ist } X \}.$$

Klar: $Q_\Delta(\mathbb{D}) = Q_{\Delta,\text{leicht}}(\mathbb{D}) \cup Q_{\Delta,\text{schwer}}(\mathbb{D})$

und $|Q_\Delta(\mathbb{D})| = |Q_{\Delta,\text{leicht}}(\mathbb{D})| + |Q_{\Delta,\text{schwer}}(\mathbb{D})|$.

Behauptung 1: $|Q_{\Delta,\text{leicht}}(\mathbb{D})| \leq (N^{\mathbb{D}})^{3/2}$

Beweis: Für jeden leichten Knoten $a \in \text{adom}(\mathbb{D})$ ist $\text{aus-Grad}^{\mathbb{D}}(a) < \sqrt{N^{\mathbb{D}}}$, und es gilt:

$$\begin{aligned}
&|\{(b,c) : (a,b,c) \in Q_\Delta(\mathbb{D})\}| \\
&\leq |\{(b,c) : (a,b) \in E^{\mathbb{D}} \text{ und } (c,a) \in E^{\mathbb{D}}\}| \\
&\leq \text{aus-Grad}^{\mathbb{D}}(a) \cdot \text{ein-Grad}^{\mathbb{D}}(a), \quad \text{wobei} \\
&\text{ein-Grad}^{\mathbb{D}}(a) := |\{w : (w,a) \in E^{\mathbb{D}}\}| = |E^{\mathbb{D}}(*,a)| \\
&\text{für } E^{\mathbb{D}}(*,a) := \{w : (w,a) \in E^{\mathbb{D}}\}.
\end{aligned}$$

Insgesamt gilt:

$$\begin{aligned}
|Q_{\Delta,\text{leicht}}(\mathbb{D})| &\leq \sum_{\substack{a \in \text{adom}(\mathbb{D}),\\ a \text{ leicht}}} |\{(b,c) : (a,b,c) \in Q_\Delta(\mathbb{D})\}| \\
&\leq \sum_{\substack{a \in \text{adom}(\mathbb{D}),\\ a \text{ leicht}}} \text{aus-Grad}^{\mathbb{D}}(a) \cdot \text{ein-Grad}^{\mathbb{D}}(a) \\
&\leq \sqrt{N^{\mathbb{D}}} \cdot \sum_{a \in \text{adom}(\mathbb{D})} \text{ein-Grad}^{\mathbb{D}}(a) \\
&\leq \sqrt{N^{\mathbb{D}}} \cdot |E^{\mathbb{D}}| \\
&= \sqrt{N^{\mathbb{D}}} \cdot N^{\mathbb{D}} = (N^{\mathbb{D}})^{3/2}
\end{aligned}$$

□ Beh 1.

Sei S^D die Menge aller schweren Knoten von D,
d.h. $S^D := \{ v \in adom(D) : v \text{ ist schwer} \}$.

Behauptung 2: $|Q_{\Delta,\text{schwer}}(D)| \leq |S^D| \cdot N^D$

Beweis:

$$\begin{aligned} Q_{\Delta,\text{schwer}}(D) &= \{ (a,b,c) : a \in S^D \text{ und } (a,b,c) \in Q_\Delta(D) \} \\ &\subseteq \{ (a,b,c) : a \in S^D \text{ und } (b,c) \in E^D \} \\ &= S^D * E^D \end{aligned}$$

Somit: $|Q_{\Delta,\text{schwer}}(D)| \leq |S^D * E^D| = |S^D| \cdot |E^D| = |S^D| \cdot N^D$

□ Beh 2

Behauptung 3: $|S^D| \leq \sqrt{N^D}$

Beweis: Für jedes $v \in S^D$ ist $|E^D(v,*)| = \text{aus-Grad}^D(v) \geq \sqrt{N^D}$.

Es gilt:

$$\begin{aligned} N^D = |E^D| &\geq \Big| \bigcup_{v \in S^D} \{ (v,w) : w \in E^D(v,*) \} \Big| \\ &= \sum_{v \in S^D} |\{ (v,w) : w \in E^D(v,*) \}| \\ &= \sum_{v \in S^D} |E^D(v,*)| \\ &\geq \sum_{v \in S^D} \sqrt{N^D} \\ &= |S^D| \cdot \sqrt{N^D} \end{aligned}$$

Somit ist $|S^D| \leq \frac{N^D}{\sqrt{N^D}} = \sqrt{N^D}$.

□ Beh 3

Insgesamt gilt:

$$|Q_\Delta(D)| = |Q_{\Delta,\text{leicht}}(D)| + |Q_{\Delta,\text{schwer}}(D)|$$

$$\overset{\text{Beh 1\&2}}{\leq} (N^D)^{3/2} + |S^D| \cdot N^D$$

$$\overset{\text{Beh 3}}{\leq} (N^D)^{3/2} + \sqrt{N^D} \cdot N^D$$

$$= (N^D)^{3/2} + (N^D)^{3/2}$$

$$= 2 \cdot (N^D)^{3/2}.$$

Unser Algorithmus zur Berechnung von $Q_\Delta(D)$ geht bei Eingabe von D wie folgt vor:

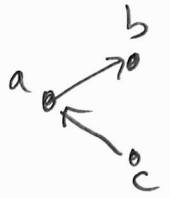
0) Initialisiere $Q_\Delta(D) := \emptyset$

1) Berechne für jedes $v \in \text{adom}(D)$ Folgendes:
die Mengen $E^D(v,*) = \{w : (v,w) \in E^D\}$ und
$E^D(*,v) = \{w : (w,v) \in E^D\}$ und
die Zahl $\text{aus-Grad}^D(v)$.
Durch geeignetes Sortieren von E^D geht das in Zeit $O(N^D \cdot \log N^D) \leq O((N^D)^{3/2})$

2) Berechne die Menge S^D aller schweren $v \in \text{adom}(D)$ und die Menge $L^D := \text{adom}(D) \setminus S^D$ aller leichten $v \in \text{adom}(D)$.
Unter Verwendung der in 1) gesammelten Infos geht das in Zeit $O(N^D)$.

3) Betrachte jedes leichte $a \in L^D$
Betrachte jedes $b \in E^D(a,*)$
Betrachte jedes $c \in E^D(*,a)$
und teste, ob $(b,c) \in E^D$.
Wenn ja, füge (a,b,c) in $Q_\Delta(D)$ ein

Skizze:



Auf die gleiche Art wie im Beweis von Beh 1 erhalten wir, dass das in Zeit $O((N^D)^{3/2} \cdot t)$ geht.

4) Betrachte jedes schwere $a \in S^D$
Betrachte jedes $(b,c) \in E^D$
und teste, ob $(a,b) \in E^D$
und
$(c,a) \in E^D$.

Skizze:

Wenn ja, füge (a,b,c) in $Q_\Delta(D)$ ein.

Das geht in Zeit $|S^D| \cdot N^D \cdot 2t = O((N^D)^{3/2} \cdot t)$.

Insgesamt berechnen wir so $Q_\Delta(D)$ in Zeit $O((N^D)^{3/2} \cdot t)$.

□ Satz Δ

Bemerkung Δ:

Der Algorithmus aus Satz Δ ist im folgenden Sinn Worst-case-optimal:

Es gibt eine Folge von σ-Datenbanken $D_1, D_2, D_3, \ldots$ so dass für $N_i := |E^{D_i}|$ für $i \in \mathbb{N}_{\geq 1}$ gilt:

$N_1 < N_2 < N_3 < \ldots$ und

$|Q_\Delta(D_i)| = (N_i)^{3/2}$.

Dazu wähle für $i \geq 1$ die DB D_i mit $E^{D_i} = [1,i] \times [1,i]$.

Dann ist $N_i = i^2$ und $Q_\Delta(D_i) = [1,i]^3$, also

$|Q_\Delta(D_i)| = i^3 = (N_i)^{3/2}$.

Literaturhinweis:

Der hier präsentierte Beweis von Satz △ ist aus der folgenden Arbeit entnommen:

"Worst-case optimal join algorithms" von Ngo, Porat, Ré, Rudra, In Proc. PODS 2012, pp. 37–48

Die Beweisidee wird dort Loomis und Whitney zugeschrieben ("An inequality related to the isoperimetric inequality" von Loomis, Whitney. In Bull. Amer. Math. Soc., 55: 961–962, 1949).