Datenbanksysteme

Wintersemester 2015/16

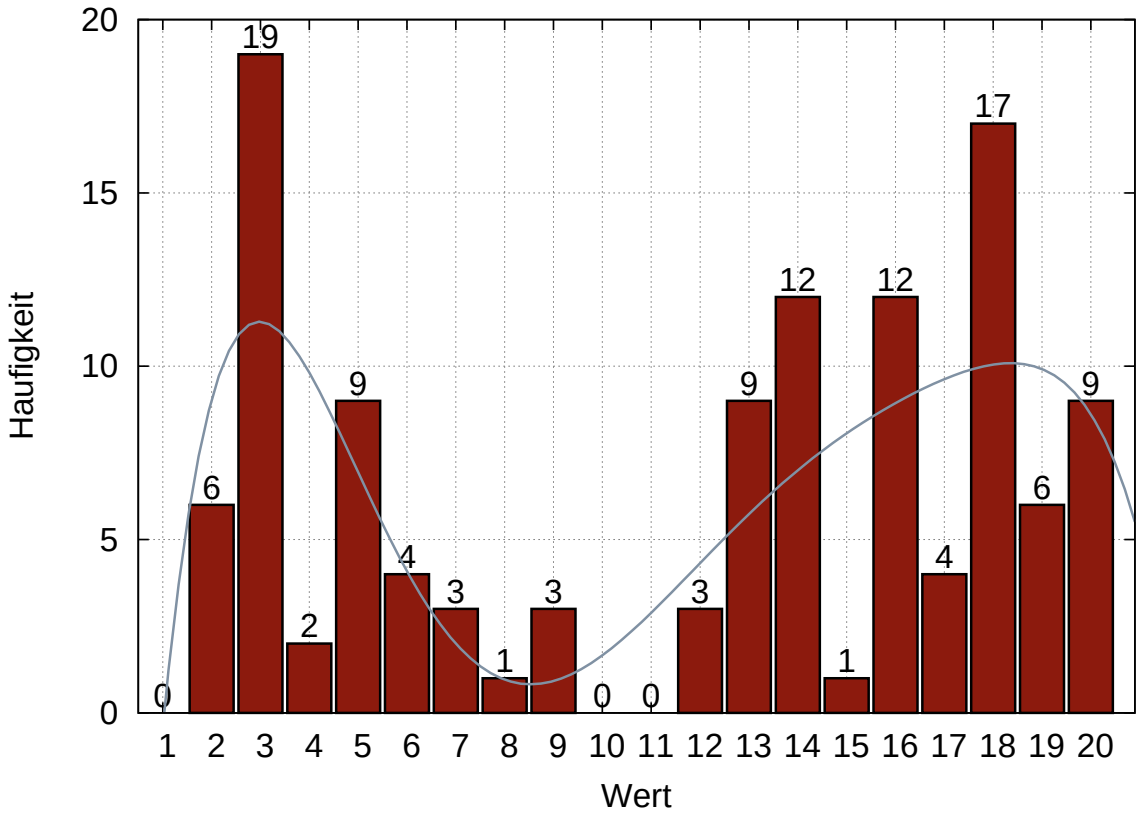
Prof. Dr.-Ing. Sebastian Michel
TU Kaiserslautern

smichel@cs.uni-kl.de

Beispieldaten

{2, 5, 3, 20, 18, 7, 16, 18, 18, 2, 2, 17, 14, 3, 20, 3, 16, 6, 7, 3, 16, 16, 15, 5, 20, 13, 16, 20, 12, 14, 13, 3, 14, 18, 14, 14, 16, 18, 19, 3, 5, 2, 5, 14, 20, 17, 3, 17, 16, 3, 2, 19, 3, 9, 13, 4, 3, 16, 14, 13, 13, 16, 20, 14, 4, 2, 3, 18, 7, 3, 5, 3, 6, 9, 18, 3, 16, 18, 20, 18, 5, 18, 5, 18, 13, 14, 19, 13, 14, 3, 14, 18, 14, 18, 18, 16, 19, 5, 3, 17, 18, 3, 19, 3, 20, 9, 16, 12, 20, 8, 12, 13, 13, 19, 18, 6, 3, 5, 18, 6}

Wiederholung: Parametrisierte Verteilungen

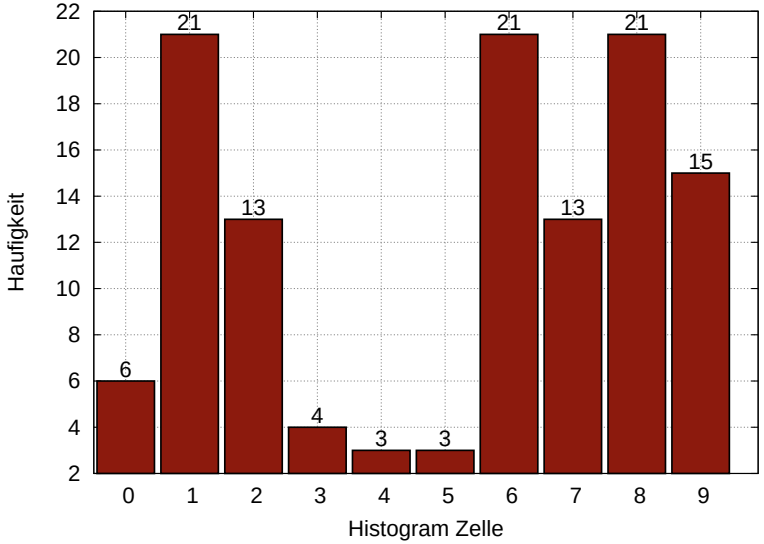


Hier, fit durch Polynom 6. Grades (mit Hilfe des Tools xmgrace):

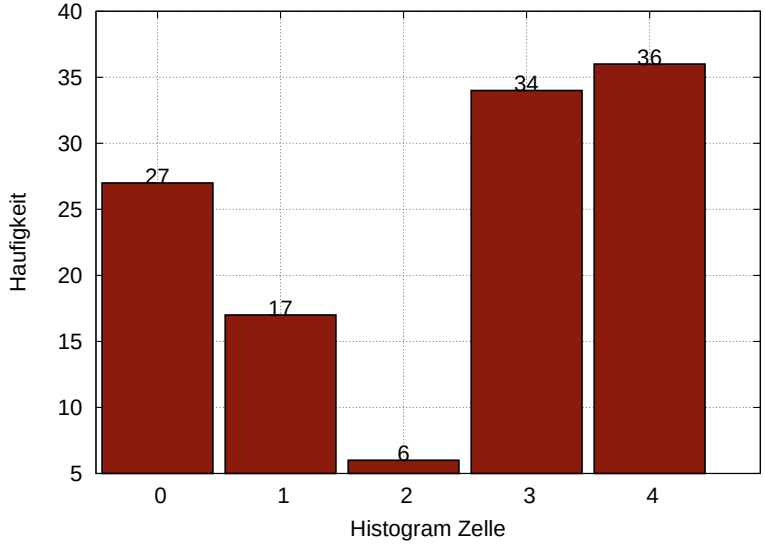
$$f(x) := -21.884 + 29.533 * x - 9.2268 * x^2 + 1.2529 * x^3 - 0.085019 * x^4$$

$$+0.0028667 * x^5 - 3.8485 * 10^{-5} * x^6$$

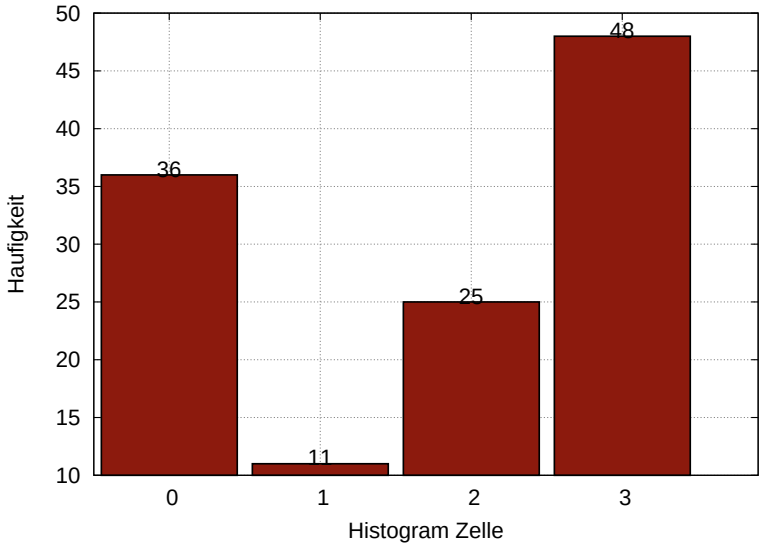
Wiederholung: Equi-Width-Histogramme



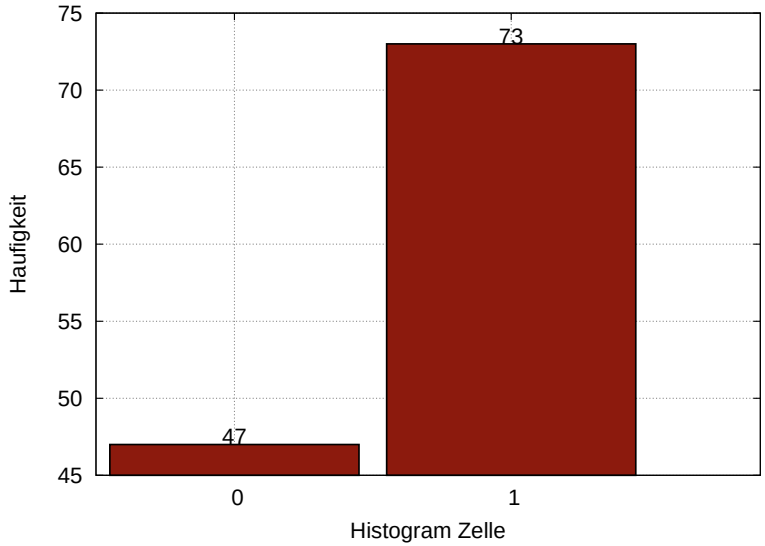
(a) Width = 2



(b) Width = 4



(a) Width = 5



(b) Width = 10

Formale Definition

Gegeben eine Menge von von n Datenpunkten s_i mit einer Zugeordneten Frequenz (Häufigkeit) $f(s_i)$. Wir schreiben auch f_i für $f(s_i)$.
Diese s_i seien bereits geordnet (obdA):

$$s_1 < s_2 < s_3 < \ldots s_n$$

Der Vektor der Häufigkeiten

$$F = [f(s_1), f(s_2), \ldots, f(s_n)]$$

Histogramm

- Ein Histogramm partitioniert diesen Vektor der Häufigkeiten in B Buckets (Zellen oder Intervalle) I_i. Dabei ist $B << n$.
- Jedes Intervall I_i wird durch einen Wert h_i (z.B. Durchschnitt) repräsentiert.

Fehler

Das heisst, für ein Intervall $I_i = [b_i,e_i]$, wobei b_i der Beginn und e_i das Ende des Intervalls markieren, wird die Häufigkeit der Werte $s_{b_i}, s_{b_i+1}, s_{b_i+2}, \ldots s_{e_i}$, die in dem Intervall enthalten sind, durch h_i approximiert.

Beispiel:

- Wie häufig kommt der Wert s_{b_i+2} vor?
- Natürlich genau $f(s_{b_i+2}) = f_{b_i+2}$ mal.
- Aber durch Histogramm eben approximiert durch h_i

Fehler

- Der Fehler, der dadurch für Wert s_{b_i+2} entsteht ist also $h_i - f_{b_i+2}$
- Normalerweise wird $|h_i - f_{b_i+2}|$ oder $(h_i - f_{b_i+2})^2$ benutzt

Sum Squared Error (SSE)

Es ist üblich h_i als Durchschnitt über die in I_i enthaltenen Daten (Häufigkeiten) zu berechnen, also

$$h_i = AVG([b_i, e_i]) = \frac{\sum_{b_i \leq k \leq e_i} F[k]}{e_i - b_i + 1}$$

Fehlermaß

Sum Squared Error (SSE)

$$SSE([a,b]) = \sum_{k=a}^{b} (F[k] - AVG([a,b]))^2$$

- Klar: **Je kleiner dieser Fehler, desto besser die Approximation** der (aller) Daten durch das Histogramm

V-Optimale Histogramme

Algorithmus zum Berechnen des bzgl. SSE optimalen Histogramms (mit Kosten SSE^*) unter Verwendung von B Intervallen (Buckets).

$$SSE^*(i,k) = min_{1 \leq j \leq i}\{SSE^*(j,k-1) + SSE([j+1,i])\}$$

für die ersten i Werte, unter Verwendung von k Buckets.
Ausnutzen, dass für beliebige i,j,k mit $0 \leq i \leq k < j \leq N$

$$SSE([i,j]) \geq SSE([i,k]) + SSE([k+1,j])$$

Literatur: Optimal Histograms with Quality Guarantees. H.V. Jagadish et al., VLDB Conference, 1998.

V-Optimale Histogramme: Erläuterung zur Berechnungsvorschrift

Berechnungsvorschrift des optimalen Fehlers für i Datenpunkte und k Buckets.

$$SSE^*(i,k) = min_{1 \leq j \leq i}\{SSE^*(j,k-1) + SSE([j+1,i])\}$$

Bedeutung

- Betrachte alle bisherigen Lösungen $SSE^*(j,k-1)$ für $k-1$ Buckets und $j \leq i$ Datenpunkte
- Nimm Fehler dieser kleineren Lösung und addiere dazu den SSE Fehler des zusätzlichen k-ten Buckets mit den Grenzen $[j+1,i]$
- Suche minimalen so berechneten Fehler für Lösung mit k Buckets und i Datenpunkten.

V-Optimale Histogramme: Algorithmus

$SSE^*(i,k)$ beschreibt optimalen Fehler für Histogram über ersten i Werten und k Zellen.

Dynamische Programmierung über

$$SSE^*(i,k) = min_{1 \leq j \leq i}\{SSE^*(j,k-1) + SSE([j+1,i])\}$$

for $k := 1$ **to** B
  for $i := 1$ **to** n
    for $j := 1$ **to** i
      if $besterror[j][k-1] + SSE(j+1,i) < besterror[i][k]$
        update $besterror[i][k]$
      ...
    end
  end
end

Initialisierung von $besterror[][]$ und Randfälle nicht im Code gezeigt.

Komplexität des Algorithmus ist $O(B\ n^2)$

Implementierungs Trick für SSE

Lemma 1 im original Papier von Jagadish et al.

$$SSE([i,j]) = \sum_{i \leq k \leq j} F[k]^2 - (j - i + 1) * AVG([i,j])^2$$

mit

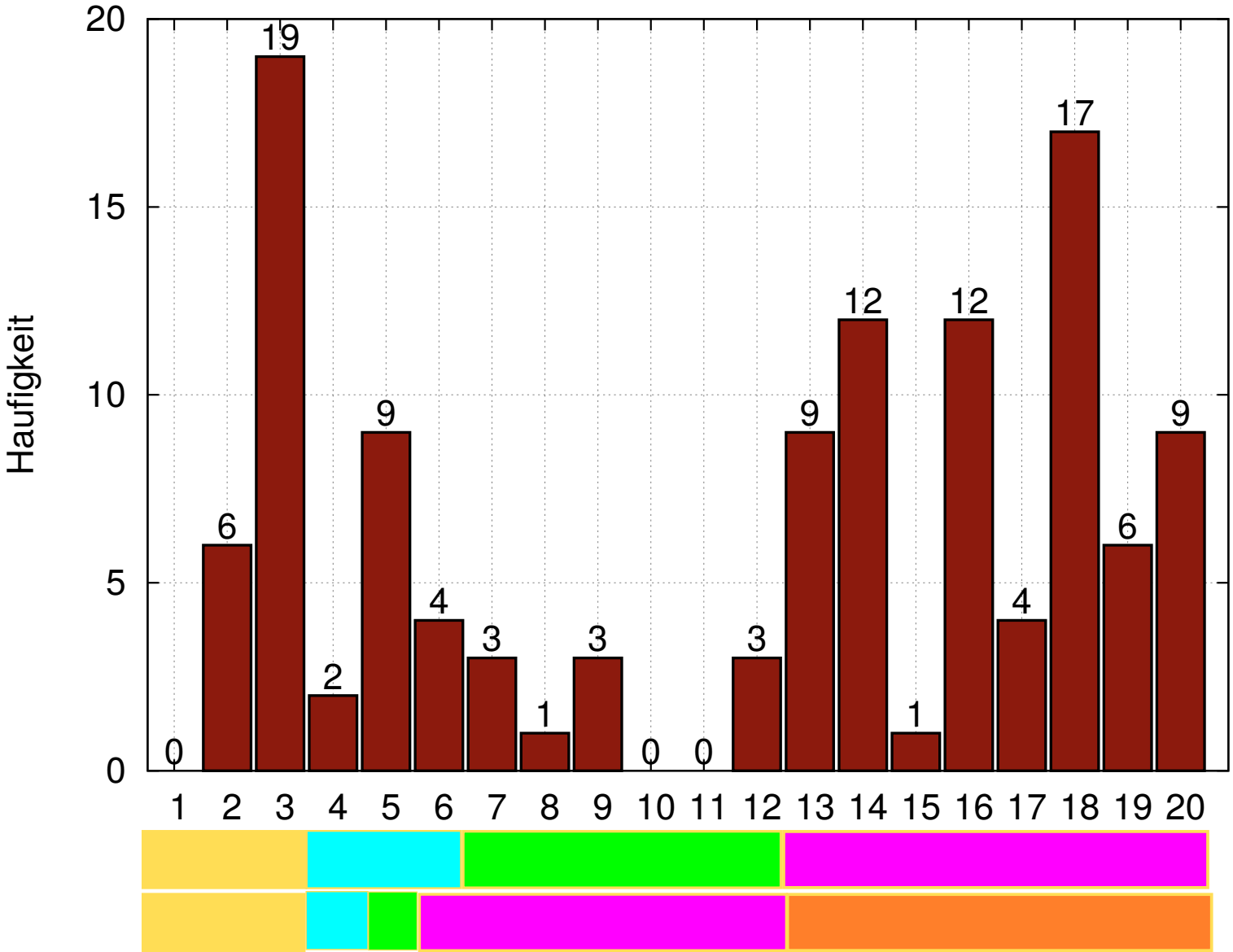
$$\sum_{i \leq k \leq j} F[k]^2 = PP[k] - PP[i-1]$$

und

$$AVG([i,j]) = \frac{P[j] - P[i-1]}{(j - i + 1)}$$

wobei $P[i] = \sum_{1 \leq k \leq i} F[k]$ und $PP[i] = \sum_{1 \leq k \leq i} F[k]^2$ (welche vorberechnet werden).

Beispiel: Für o.g. Daten und B = 4 bzw. 5 Zellen



Schätzung mit Histogrammen

- Obwohl man für diskrete Daten auch Punktanfragen bearbeiten kann, ist dies im kontinuierlichen Fall nicht möglich.
- Es liegen unendlich viele Werte in jeder Histogrammzelle (mit Häufigkeit 0)
- D.h. es kann im kontinuierlichen Fall "nur" nach Häufigkeiten für Intervalle gefragt werden.

Fehlermaße

- Ist für einen exakten Wert x eine Schätzung (Näherungswert) $\hat{x}$ gegeben,
  - so heisst $|\hat{x} - x|$ absoluter Fehler und
  - $\frac{|\hat{x} - x|}{x}$ im Fall $x \neq 0$ relativer Fehler
- Für mehrere solcher Beobachtungen: Sum Squared Error (SSE), Mean Absolute Error (MAE), Mean Squared Error (MSE)

Deskriptive Statistik

- Minimaler und maximaler Wert eines Attributs
- Durchschnittlicher Wert und Median
- Weitere Aussagen über Verteilungen
- Beispielwerte

Anwendungen

- Produkt-Mananger: "In 99,9% aller Fälle liegt die Antwortzeit unseres Systems XYZ unter 10ms."
- Professor: "75% aller Studenten, die die Klausur mitgeschrieben haben, haben mindestens 80 Punkte erreicht."

Quantile einer Verteilung

Definition

Gegeben eine Zufallsvariable X mit Verteilungsfunktion F. Für ein $p \in [0,1]$, definieren wir $F^{-1}(p)$ als kleinsten Wert x, so dass $F(x) \geq p$.

Dieser Wert $F^{-1}(x)$ wird das p Quantil von X genannt. Die Funktion F^{-1} wird Quantilfunktion genannt.

- Das p Quantil wird auch $100p$ Perzentil genannt.
- Das $1/2$ Quantil, bzw. das 50. Perzentil einer Verteilung wird auch Median genannt.
- Das $1/4$ Quantil, bzw. das 25. Perzentil, wird auch **erstes Quartil** genannt,
- das $3/4$ Quantil, bzw. das 75. Perzentil, wird auch **drittes Quartil** genannt.

Probabilistic Counting

$V(R,A)$ ist die Anzahl der verschiedenen Attributsausprägungen für Attribut A.
Wie kann diese Größe berechnet werden?

- Klar, via Duplikat-Eliminierung durch Sortieren oder durch Hashing
- Oder durch probabilistische Methoden (Schätzer)

Flajolet Martin (FM) Sketch (aka. Hash Sketch)

Vorgeschlagen von Flajolet und Martin in 1985[1]

- Erzeuge einen leeren Bitvektor B der Länge $m = log(N)$
- Scan über Eingabedaten: Dabei wird für jedes Objekt eine Position im Bitvektor berechnet und auf "1" gesetzt:
  - Hashing eines Objekts i in eine m-bit Zahl $h(i)$
  - Berechne Position k des am wenigsten signifikanten "1" Bits von h(i)
  - Setze bit $B[k]$ auf "1"

Beispiel

Eingabe: 17, 5, 19, 211, 17, 5, 31
Annahme h(17)=010100, dann ist das am wenigsten signif. 1 Bit = 2
Annahme h(5)=000101, dann ist das am wenigsten signif. 1 Bit = 0

[1] Philippe Flajolet, G. Nigel Martin: Probabilistic Counting Algorithms for Data Base Applications. J. Comput. Syst. Sci. 31(2): 182-209 (1985)

Schätzer

- Am Ende sieht B dann z.B. so aus: $B = 111010$
- Betrachte die Position t des am weitesten links stehenden "0" Bits, hier im Beispiel $t = 3$.
- Dann ergibt sich die Schäzung für die tatsächliche Anzahl n als

$$\hat{n} = 2^t \times 0.7735$$

in unserem Beispiel mit $t = 3$: $\hat{n} = 2^3 \times 0.7735 \approx 6.188$

Verbesserung der Schätzung

Verwendung mehrerer Bitvektoren B (entsprechend mit unterschiedlichen Hashfunktionen h) und Berechnung eines durchschnittlichen Werts von t.

Idee/Intuition

- $B[0]$ wird ungefähr $n/2$ mal gesetzt
- $B[1]$ wird ungefähr $n/4$ mal gesetzt

...

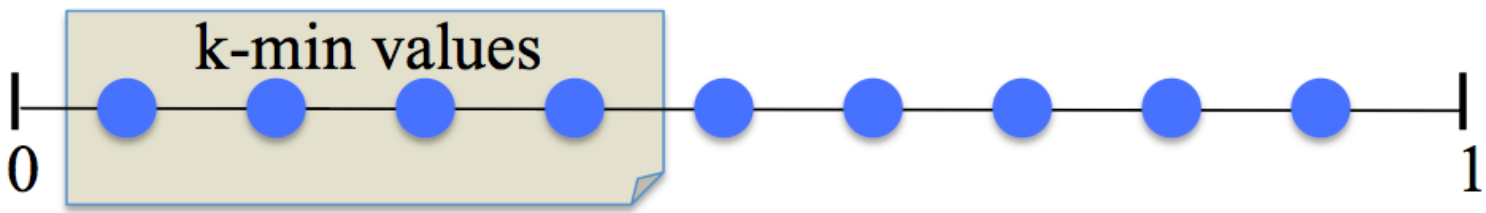
Also:

- $B[i] = 0$ falls $i >> log_2(n)$
- $B[i] = 1$ falls $i << log_2(n)$
- "Mischung" aus 1s und 0s um $i \approx log_2(n)$ herum

K-Min Value (KMV) Sketch

Problemstellung wie zuvor: Gegeben eine Multimenge S von Elementen, finde die Anzahl n der unterschiedlichen (distinct) Elemente.

- Wende Hashfunktion auf Elemente an zur gleichverteilten Abbildung auf $[0,1]$
- Die KMV Synopse besteht dann aus den k kleinsten dieser Werte $L = \{U_{(1)}, U_{(2)}, \ldots, U_{(k)}\}$

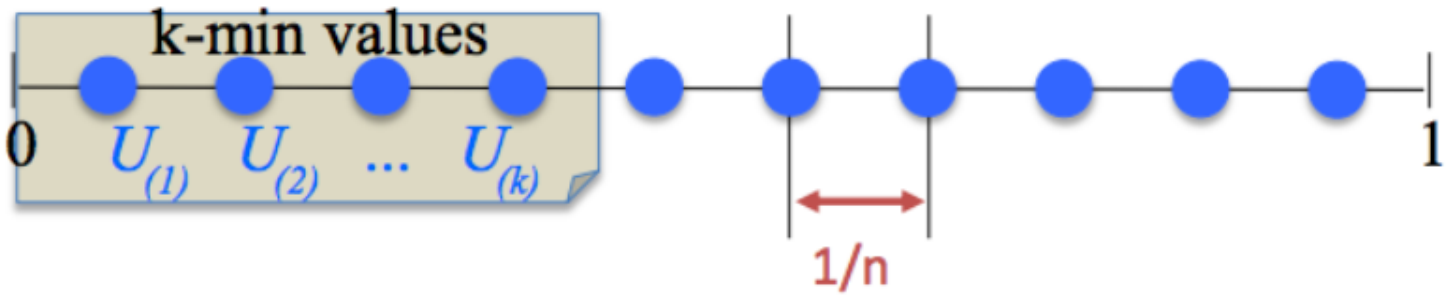


- Unbiased (Deutsch: erwartungstreuer) Schätzer erhalten wir durch

$$n^{UB} = (k-1)/U_{(k)}$$

K-Min Value (KMV) Sketch (2)

Wieso funktioniert das?



- $L = \{U_{(1)}, U_{(2)}, \ldots, U_{(k)}\}$
- Distanz zwischen den Hashwerten ist $1/n$, ausgehend von Gleichverteilung.
- Wir möchten n abschätzen. Welche Größe kennen wir?
- Beobachtung

$$E[U_{(k)}] = k \times \frac{1}{n} \qquad \text{also} \qquad \hat{n}^{BE} = \frac{k}{U_{(k)}}$$

$\hat{n}^{BE}$ ist biased (nicht erwartungstreu).

K-Min Value (KMV) Sketch (3)

Beispiel

- MD5 als Hashfunktion, Abbildung auf $[0,1]$
- Hier, Elemente als einfache Buchstaben a–z.
- Die sortierten Hashwerte der 26 Buchstaben sind [0.043, 0.172, 0.281, 0.354, 0.382, 0.421, 0.443, 0.459, 0.463, 0.523, 0.556, 0.565, 0.569, 0.57, 0.59, 0.644, 0.652, 0.675, 0.682, 0.724, 0.818, 0.864, 0.89, 0.938, 0.994, 0.997]

	$\hat{n}^{UB}$	$U_{(k)}$
1	0.000	0.043
2	5.814	0.172
3	7.117	0.281
4	8.475	0.354
5	10.471	0.382
6	11.876	0.421
7	13.544	0.443
8	15.251	0.459
9	17.279	0.463
10	17.208	0.523
11	17.986	0.556
12	19.469	0.565
13	21.090	0.569
14	22.807	0.570
15	23.729	0.590
16	23.292	0.644
17	24.540	0.652
18	25.185	0.675
19	26.393	0.682

Allgemeine Beobachtungen

- Warum werden nur distinct Elemente gezählt? Abbildung von Duplikaten auf gleichen Wert.
- Gegeben zwei KMV Synopsen, für Multimengen A und B, kann man eine Synopse angeben für $A \cup B$? Ja, einfach kleinste der $2 * k$ Werte nehmen.

Tuning von Datenbanken

- Statistiken (Histogramme, etc.) müssen explizit angelegt werden
- Andernfalls liefern die Kostenmodelle falsche Werte
- Oracle:
  - `analyze table Professoren compute statistics for table;`
  - Man kann sich auch auf approximative Statistiken verlassen
  - Anstatt compute verwendet man estimate
- DB2:
  - `runstats on table ....`
- Postgres:
  - `analyze`
  - `http://www.postgresql.org/docs/9.0/static/catalog-pg-statistic.html`

Statistiken in Postgresql

Tabelle analysieren mit:

```
analyze lineitem;
```

Daten werden in einer internen Tabelle (pg_statistic) abgelegt; pg_stats ist eine (besser zu lesende) Sicht darauf.

```
select *
from pg_stats
where tablename = 'lineitem';
```

Beispiel: Auszug aus pg_stats (in Postgresql)

Für Tabelle `lineitem` des TPC-H Datasets (`http://www.tpc.org/tpch/`)

attname name	inheri boole	null_frac real	avg_width integer	n_distinct real	most_common_vals anyarray	most_common_freqs real[]	histogram_bounds anyarray	correlation real
l_orderkey	f	0	4	396485	{73190,578534,	{0.0001,0.0001,	{3,64448,1278	1
l_partkey	f	0	4	189666	{5893,21347,14	{0.000133333,0.	{2,2005,4000,5	0.003532
l_suppkey	f	0	4	10018	{9118,7099,747	{0.0004,0.00036	{2,96,200,303,	0.007977
l_linenumber	f	0	4	7	{1,2,3,4,5,6,7	{0.2474,0.21653		0.178991
l_quantity	f	0	5	50	{41.00,28.00,4	{0.0219,0.02163		0.022648
l_extendedprice	f	0	8	-0.12923	{73119.15,1216	{0.000133333,0.	{930.00,1474.4	0.004723
l_discount	f	0	4	11	{0.08,0.09,0.0	{0.0952,0.0922,		0.083595
l_tax	f	0	4	9	{0.07,0.06,0.0	{0.116433,0.115		0.104229
l_returnflag	f	0	2	3	{N,A,R}	{0.508433,0.247		0.371031
l_linestatus	f	0	2	2	{O,F}	{0.502267,0.497		0.499684
l_shipdate	f	0	4	2510	{1993-04-21,19	{0.000833333,0.	{1992-01-06,19	-0.00355
l_commitdate	f	0	4	2458	{1993-01-31,19	{0.0008,0.0008,	{1992-02-04,19	-0.00370
l_receiptdate	f	0	4	2522	{1994-03-23,19	{0.000966667,0.	{1992-01-10,19	-0.00361
l_shipinstruct	f	0	26	4	{"DELIVER IN PI	{0.2548,0.25283		0.241248
l_shipmode	f	0	11	7	{"SHIP ",	{0.1479,0.14653		0.137974
l_comment	f	0	27	0.153266	{" furiously",	{0.000233333,0.	{"about the ac	0.012145

Erläuterung zu pg_stats

http://www.postgresql.org/docs/9.2/static/view-pg-stats.html

- **null_frac**: Fraction of column entries that are null
- **n_distinct**: If greater than zero, the estimated number of distinct values in the column. If less than ...
- **most_common_vals**: A list of the most common values in the column. (Null if no values seem to be more common than any others.)
- **most_common_freqs**: A list of the frequencies of the most common values, i.e., number of occurrences of each divided by total number of rows. (Null when most_common_vals is.)

Erläuterung zu pg_stats (2)

http://www.postgresql.org/docs/9.2/static/view-pg-stats.html

- **histogram_bounds**: A list of values that divide the column's values into groups of approximately equal population. The values in most_common_vals, if present, are omitted from this histogram calculation.
- **correlation**: Statistical correlation between physical row ordering and logical ordering of the column values. This ranges from -1 to +1. When the value is near -1 or +1, an index scan on the column will be estimated to be cheaper than when it is near zero, due to reduction of random access to the disk.

Beobachtungen

Korrelation zwischen physischer und logischer Speicherung

- l_orderkey hat Korrelationswert von 1 (!)
- Was bedeutet dies bzw. könnte bedeuten?

Frequent Values und Distinct Values

- l_shipinstruct ist "DELIVER IN PERSON" in ca. 25% aller Tupel.
- Es gibt ohnehin nur 4 unterschiedliche Werte für diese Spalte

Was bedeuten diese Beobachtungen bzgl. Notwendigkeit Indexe anzulegen bzw. evtl. vorhandene Indexe zu benutzen?

Histogramm

Für Spalte l_extendedprice in Tabelle lineitem (6001215 Tupel)

Die ersten 36 Werte aus histogram_bounds

930	9625,12	18639,95
1474,44	10529,84	19346,36
1943,94	11282,1	20020,8
2876,78	12023,41	20742,2
3540,48	12838,14	21406,91
4366,08	13624,3	22120,91
5158,89	14327,04	22799,66
5816,1	15079,54	23495,52
6600,48	15850,9	24180,42
7387,1	16550,1	24868,61
8064,56	17200,04	25609,44
8919,2	17909,52	26433

Equi-Depth-Histogramm mit 100 Zellen. Jede Histogrammzelle also beschreibt/umfasst rund 60 000 Werte.

Wie viele Tupel haben l_extendedprice < 1600?

Selektivitätsschätzung: Zusammenfassung

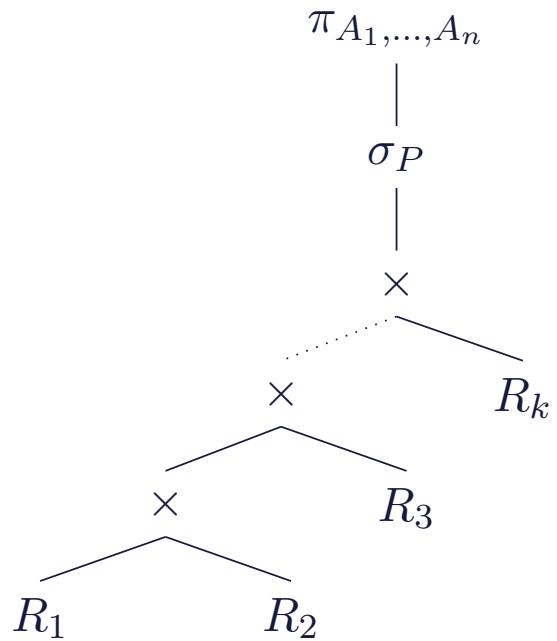
- Kosten der Operatoren hängen (neben Implementierung) von Größe der Eingaben ab
- Für einen Anfrageplan kann so die Anzahl der erwarteten Ergebnisse (Tupel) sowie die erwartete Größe der anfallenden Zwischenergebnisse berechnet werden.
- Je nach Verfügbarkeit an Statistiken können Schätzungen sehr grob oder recht genau sein, vlg. Schätzung von $|\sigma_{A<c}(R)| = 1/3 * |R|$ mit Wert von Histogrammen.

Anfrageoptimierung

Kanonische Übersetzung

select A_1, ..., A_n
from R_1, ..., R_k
where P;

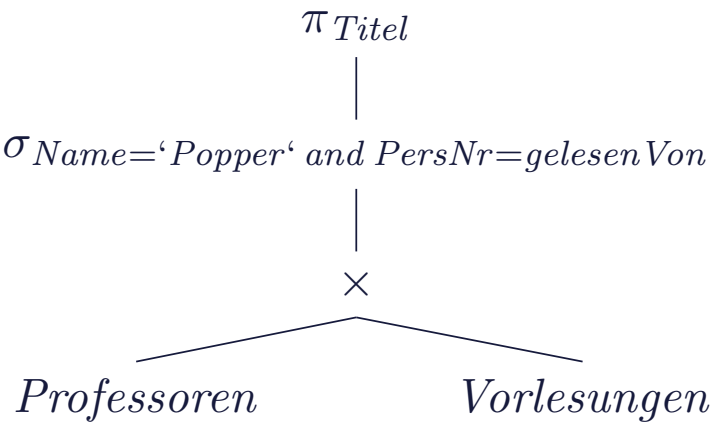
$\Rightarrow$



Kanonische Übersetzung

select Titel
from Professoren, Vorlesungen
where Name=‘Popper‘ **and**
PersNr = gelesenVon;

$\Rightarrow$

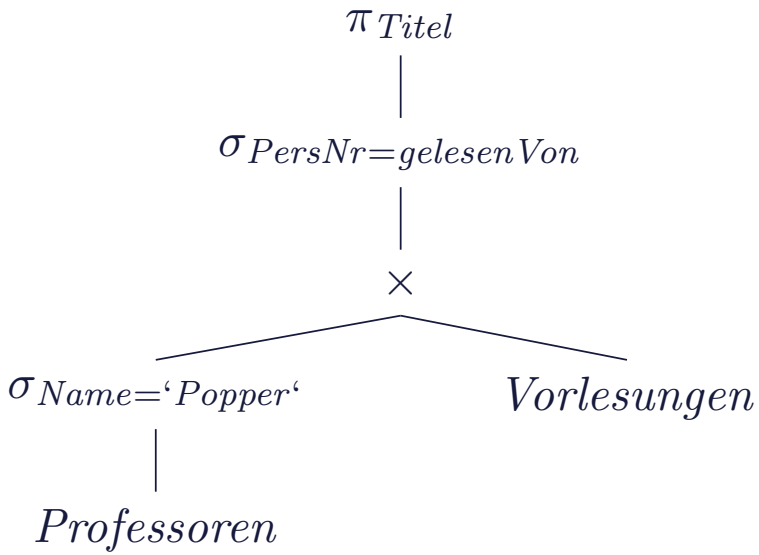


$$\pi_{Titel}(\sigma_{Name=‘Popper‘\ and\ PersNr=gelesenVon}(Professoren \times Vorlesungen))$$

Ein einfacher Optimierungschritt

select Titel
from Professoren, Vorlesungen
where Name=‘Popper‘ **and**
PersNr = gelesenVon;

$\Rightarrow$



$$\pi_{Titel}(PersNr = gelesenVon(\sigma_{Name=‘Popper‘}Professoren \times Vorlesungen))$$

Äquivalenzerhaltende Transformationsregeln

1. **Aufbrechen von Konjunktionen im Selektionsprädikat**

$$\sigma_{c_1 \wedge c_2 \wedge ... \wedge c_n}(R) \equiv \sigma_{c_1}(\sigma_{c_2}(...(\sigma_{c_n}(R))...))$$

2. σ **ist kommutativ**

$$\sigma_{c_1}(\sigma_{c_2}(R)) \equiv \sigma_{c_2}(\sigma_{c_1}(R))$$

3. π**-Kaskaden**
   Falls $L_1 \subseteq L_2 \subseteq ... \subseteq L_n$, dann gilt

$$\pi_{L_1}(\pi_{L_2}(...(\pi_{L_n}(R))...)) \equiv \pi_{L_1}(R)$$

Äquivalenzerhaltende Transformationsregeln

4. **Vertauschen von σ und π**
   Falls die Selektion sich nur auf Attribute $A_1, ..., A_n$ der Projektionsliste bezieht, können die beiden Operationen vertauscht werden:

$$\pi_{A_1,...,A_n}(\sigma_c(R)) \equiv \sigma_c(\pi_{A_1,...,A_n}(R))$$

5. **$\cup, \cap$ und $\bowtie$ sind kommutativ**

$$R \bowtie_c S \equiv S \bowtie_c R$$

Äquivalenzerhaltende Transformationsregeln

6. **Vertauschen von σ mit $\bowtie$**
   Falls das Selektionsprädikat c nur auf Attribute der Relation R zugreift, kann man die beiden Operationen vertauschen:

$$\sigma_c(R \bowtie_j S) \equiv \sigma_c(R) \bowtie_j S$$

   Falls das Selektionsprädikat c eine Konjunktion der Form $c_1 \wedge c_2$ ist und c_1 sich nur auf Attribute aus R und c_2 sich nur auf Attribute aus S bezieht, gilt folgende Äquivalenz:

$$\sigma_c(R \bowtie_j S) \equiv \sigma_{c_1}(R) \bowtie_j \sigma_{c_2}(S)$$

Äquivalenzerhaltende Transformationsregeln

7. **Vertauschen von π mit $\bowtie$**
   Die Projektionsliste L sei: $L = \{A_1, ..., A_n, B_1, ..., B_m\}$, wobei A_i Attribute aus R und B_i Attribute aus S seien. Falls sich das Joinprädikat c nur auf Attribute aus L bezieht, gilt folgende Umformung:

$$\pi_L(R \bowtie_c S) \equiv (\pi_{A_1,...,A_n}(R)) \bowtie_c (\pi_{B_1,...,B_m}(S))$$

Äquivalenzerhaltende Transformationsregeln

8. **Die Operationen $\bowtie, \cap, \cup$ sind jeweils (einzeln betrachtet) assoziativ.** Wenn also Φ eine dieser Operationen bezeichnet, so gilt:

$$(R\Phi S)\Phi T \equiv R\Phi(S\Phi T)$$

9. **Die Operation σ ist distributiv mit $\cap, \cup, -$.** Falls Φ eine dieser Operationen bezeichnet, gilt:

$$\sigma_c(R\Phi S) \equiv (\sigma_c(R))\Phi(\sigma_c(S))$$

10. **Die Operation π ist distributiv mit $\cup$.**

$$\pi_c(R \cup S) \equiv (\pi_c(R)) \cup (\pi_c(S))$$

Äquivalenzerhaltende Transformationsregeln

11. Die **Join- und/oder Selektionsprädikate** können mittels **de Morgan's Regeln** umgeformt werden

$$\neg(c_1 \wedge c_2) \equiv (\neg c_1) \vee (\neg c_2)$$

$$\neg(c_1 \vee c_2) \equiv (\neg c_1) \wedge (\neg c_2)$$

12. Ein **kartesisches Produkt**, das von einer Selektionsoperation gefolgt wird, deren Selektionsprädikat Attribute aus beiden Operanden des kartesischen Produktes enthält, kann in eine Joinoperation umgeformt werden.

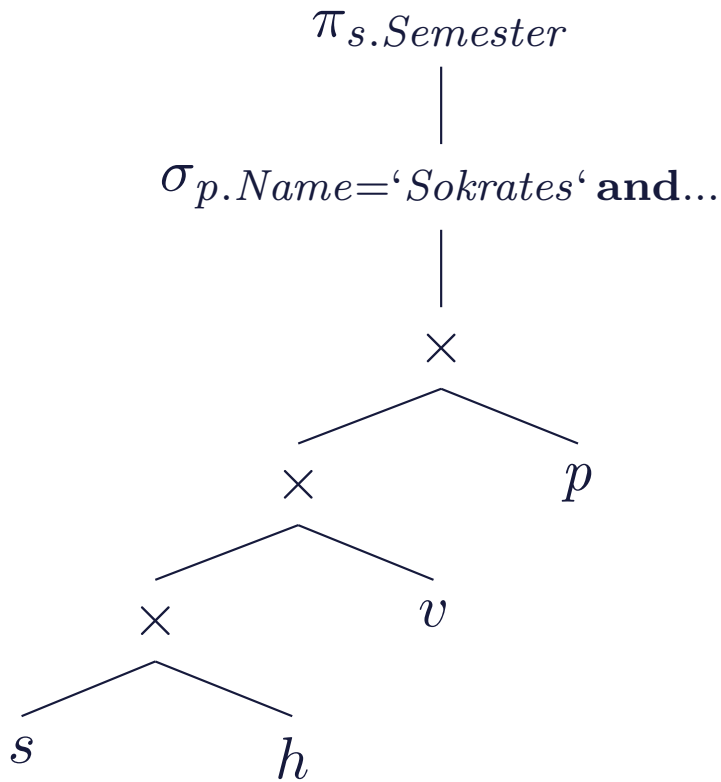
Vorgehensweise zur Optimierung

1. Aufbrechen von Selektionen
2. Verschieben der Selektionen soweit wie möglich nach unten im Operatorbaum (englisch: pushing selections)
3. Zusammenfassen von Selektionen und Kreuzprodukten zu Joins
4. Bestimmung der Reihenfolge der Joins in der Form, dass möglichst kleine Zwischenergebnisse entstehen
5. unter Umständen Einfügen von Projektionen
6. Verschieben der Projektionen soweit wie möglich nach unten im Operatorbaum

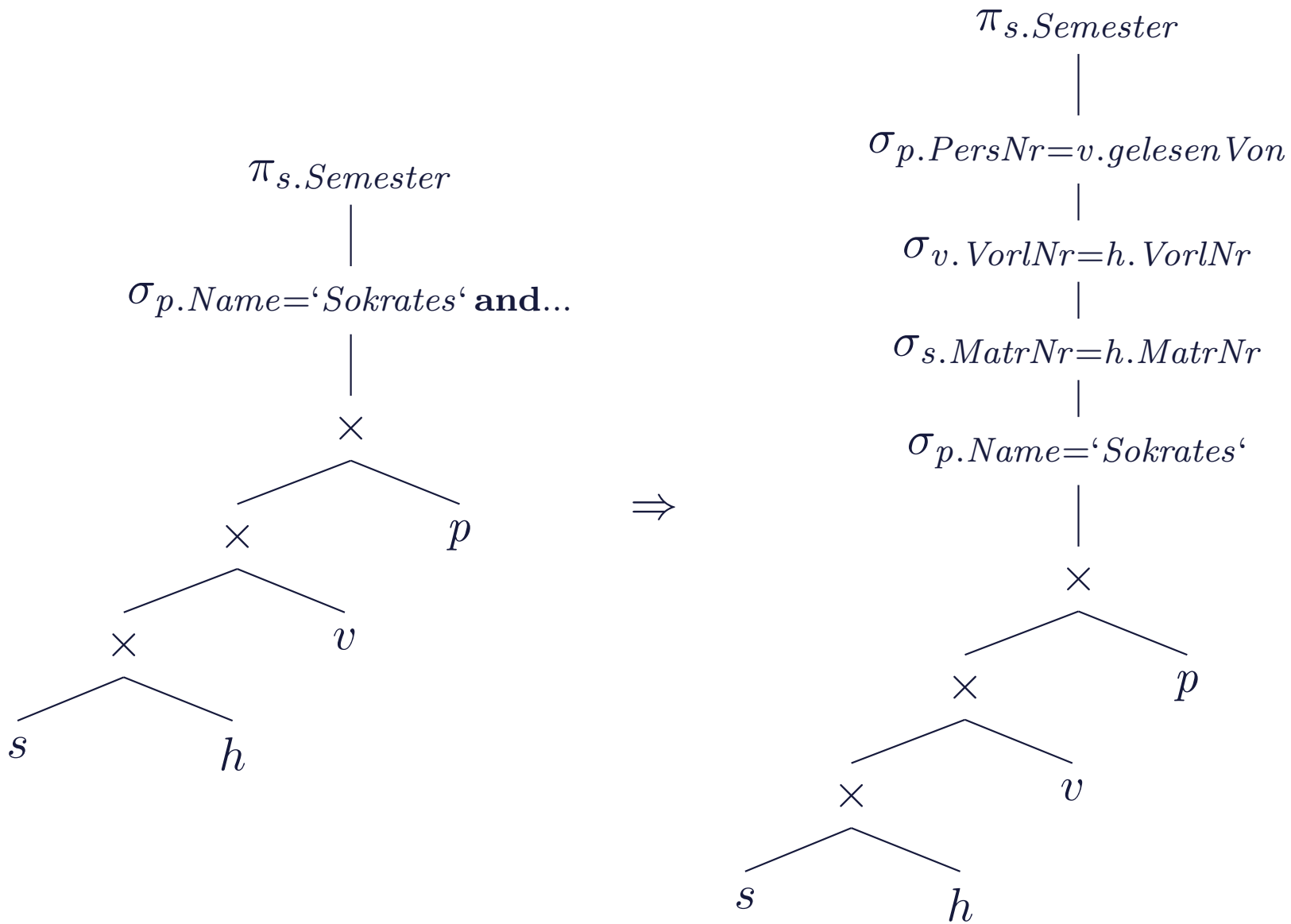
Beispiel

select distinct s.Semester
from Studenten s, hoeren h
Vorlesungen v, Professoren p
where p.Name=‘Sokrates‘ **and**
v.gelesenVon = p.PersNr **and**
v.VorlNr = h.VorlNr **and**
h.MatrNr = s.MatrNr;

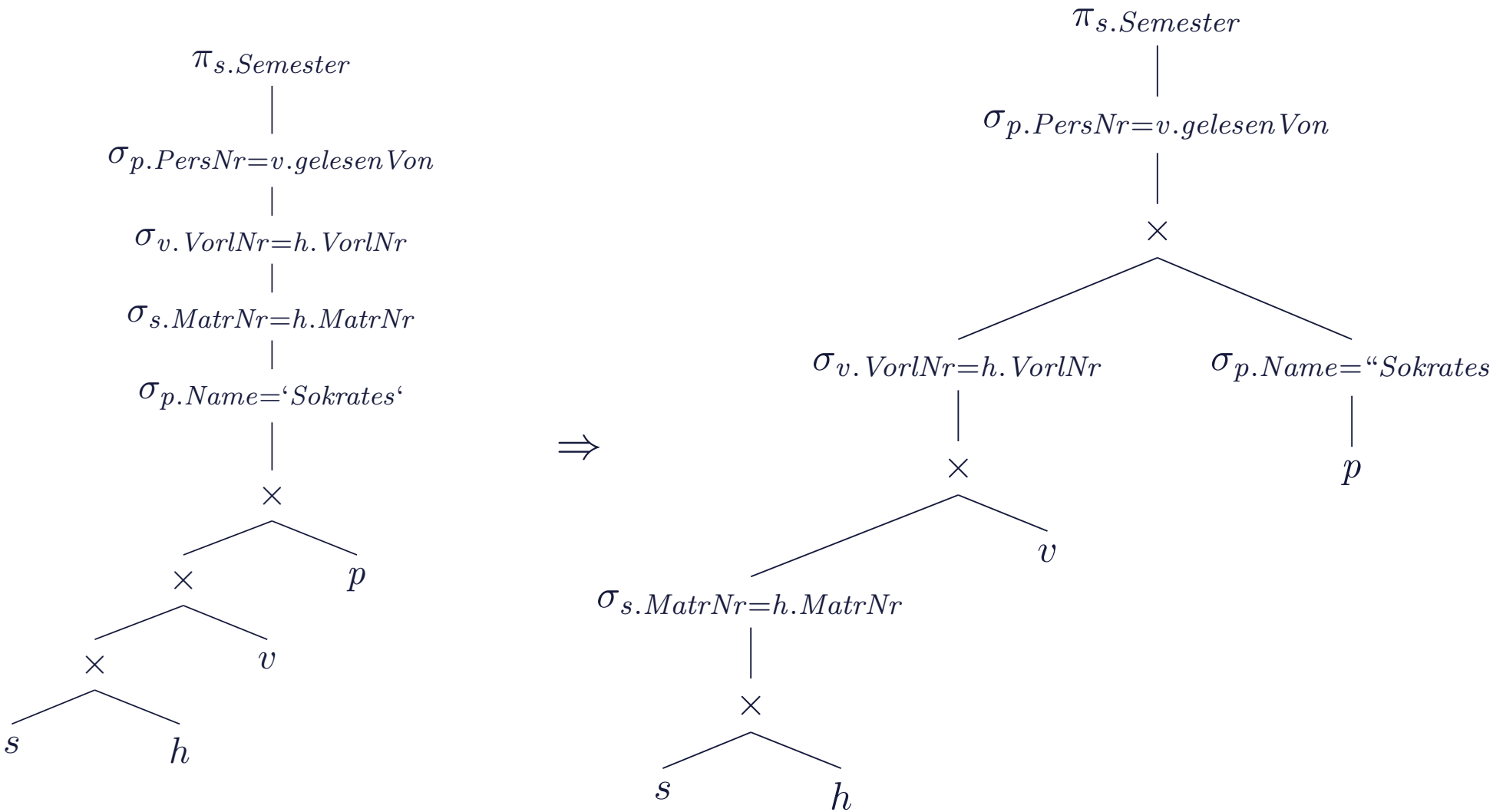
$\Rightarrow$



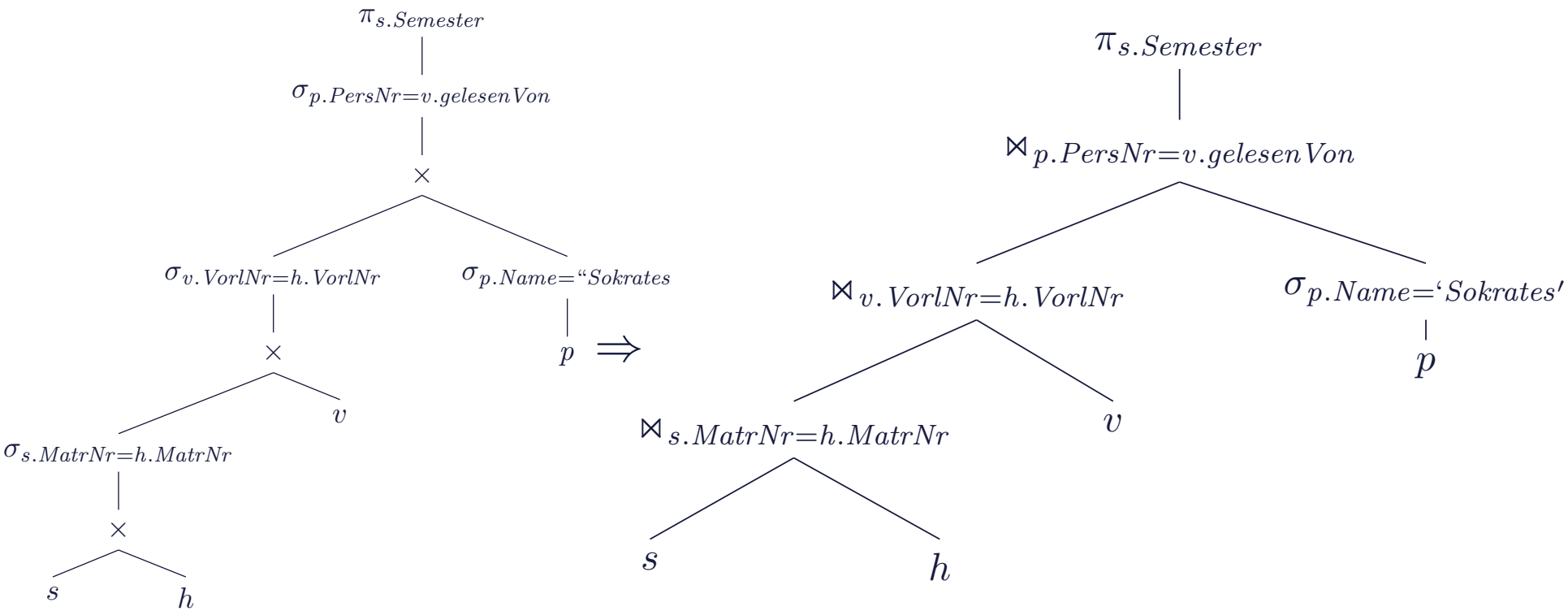
Aufspalten der Selektionsprädikate



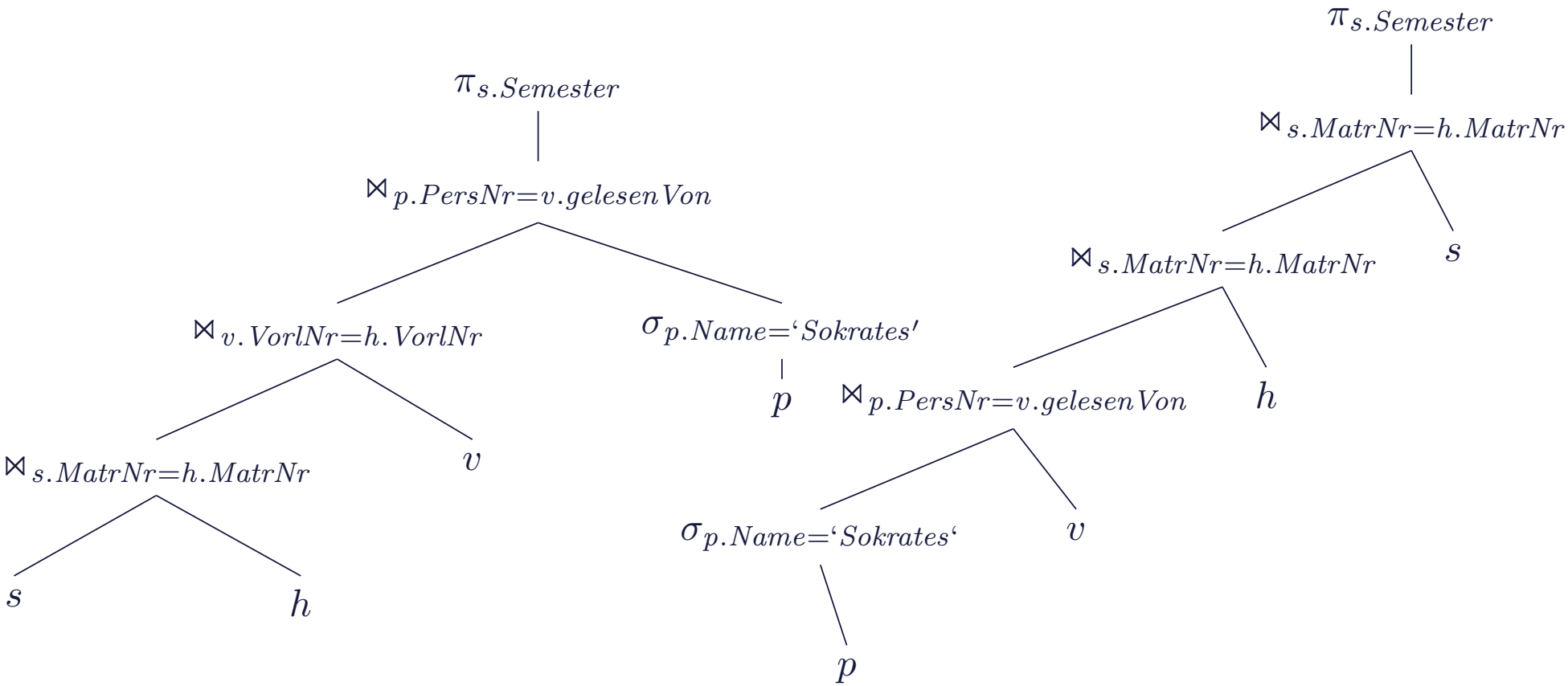
Verschieben der Selektionsprädikate



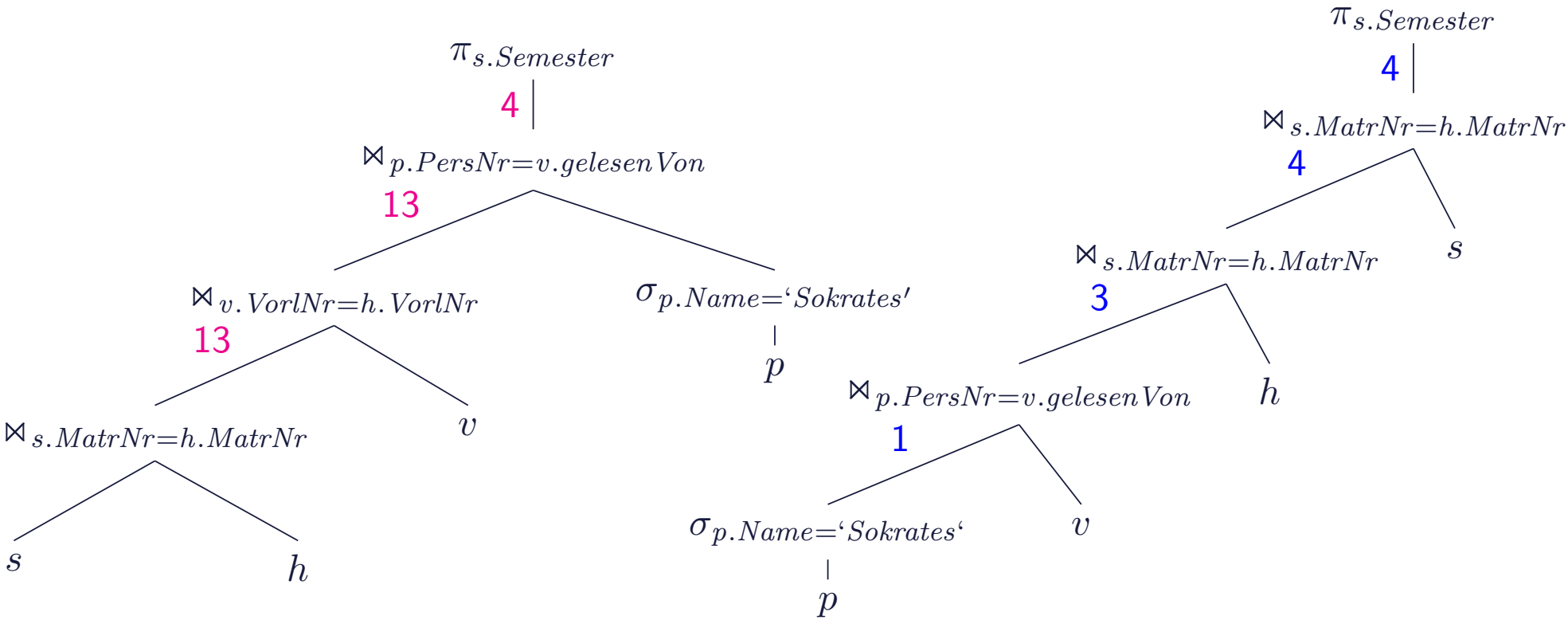
Zusammenfassung von Selektionen und Kreuzprodukten zu Joins



Optimierung der Joinreihenfolge



Effekt: Reduzierung der Zwischenergebnisse



Diese Zwischenkosten müssen natürlich geschätzt werden. Der Optimierer kann dann den günstigsten Plan bzgl. dieser geschätzten Kosten auswählen .