Entwurf und Implementation des Textarray auf Sekundärspeicher im Rahmen des XEE Projektes

Guido Draheim
<draheim@informatik.hu-berlin.de>

Dezember 2002

Einführung AST/TA

- XML Texte — mit semantischer Auszeichnung - aus Datenbanken generiert
- Mustersuche IR — Abspeichern / Strukturupdate – Kombination / Attribute
- Update DB — Längenänderung Felder – Records entfernen / einfügen
- AST/TA Trennung — vorherige Bsp. wirken nur auf je e. Teil – Verwendung herkömmlicher (IR) Algos
- Separatoren — für DB Auszeichnung und Mustersuche – div/span display vergleich

TextArray Schema

- Anforderungen — Auffinden – Weitersetzen (Pfad) – Einfügen/Löschen – Nutzungsgrad (Ausgleichen)
- B*-Tree — Mehrwegebaum – Datenblöcke in Endknoten – Füllstände (Ins/Del) - Ausgleichen
- Ausgleichen — 2-Block vs. 3-Block Ausgleichen / 2er Delete / 2er Insert / 4er Insert / 4 Delete
- Collapse — Tree-Insert / Tree-Collapse / persistant pfad / ebenen ausgleich
- Pfade — Zugriffspfade / Untebaum / Locking / NewRoot

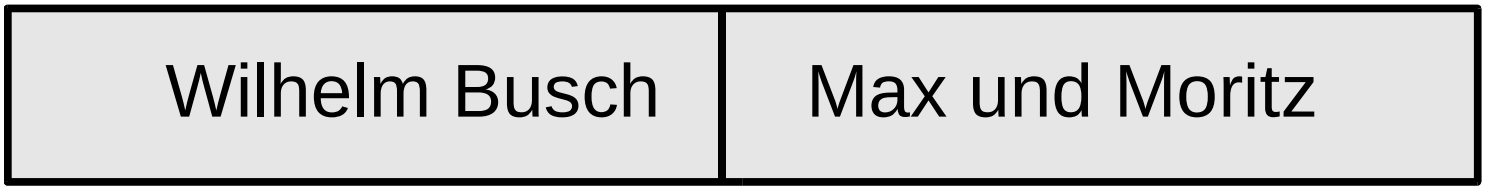
Verwendung

- Messung — 50% minimum / 66% statistisches Mittel
- Ausblick — 3er-Ausgleich / Seperatorsummen / Angepasste Ops / Multi-AST / View-Rights
- Einblick — Wiederverwendung Algos aus IR / DB – schneller / billiger / anwendbarer
- Vorabschau — PCRE Muster auf Textarray / tree copying / Speed – nächster vortrag

Wilhelm Busch schrieb Max und Moritz

(1)XML-Deklaration
(2)Dokument-Typ-Deklaration (=> DTD, *document type definition*)
(3)XML-Dokument-Instanz

```
<?xml version=“1.0“ encoding=“UTF-8“ ?>
<!DOCTYPE book SYSTEM “book.dtd“>
<book>
  <author>Wilhelm Busch</author> schrieb
  <title>Max und Moritz</title>
</book>
```



<author> <title>

<author>Wilhelm Busch</author><title>Max und Moritz</title>

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <record>
    <author>Wilhelm Busch</author>
    <title>Max und Moritz</title>
  </record>
</book>
```

Wilhelm Busch schrieb Max und Moritz

```
<?xml version=“1.0“ encoding=“UTF-8“ ?>
<!DOCTYPE book SYSTEM “book.dtd“>
<book>
  Wilhelm Busch schrieb Max und Moritz
</book>
```

.author. >> *[name] schrieb [name]* << .title.

```
<?xml version=“1.0“ encoding=“UTF-8“ ?>
<!DOCTYPE book SYSTEM “book.dtd“>
<book>
  <author>Wilhelm Busch</author> schrieb
  <title>Max und Moritz</title>
</book>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <author>Wilhelm Busch</author> schrieb
  <title>Max und Moritz</title>
  im Jahre <year>1895</year>
</book>
```

<author><title><year> .... <title published="[year]">

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <author>Wilhelm Busch</author> schrieb
  <title published="1895"
        >Max und Moritz</title>
  im Jahre <year>1895</year>
</book>
```

```
<?xml version=“1.0“ encoding=“UTF-8“ ?>
<!DOCTYPE book SYSTEM “book.dtd“>
<book>
  <author>Wilhelm Busch</author> schrieb
  <title published=“1895“
        >Max und Moritz</title>
<!-- im Jahre <year>1895</year> -->
</book>
```

When=“//title[@published='1895']“

direkt aus xml daten(bank) statt datenbank verknüpfung:

title	year
Max und Moritz	1895

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <author>Wilhelm Busch</author> schrieb
  <title published="1895"
        >Max und Moritz</title>
<!-- im Jahre <year>1895</year> -->
</book>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <author>Wilhelm Busch</author> schrieb
  <title published="1895"
        >Max und Moritz</title>
<!-- im Jahre <year>1895</year> -->
</book>
```

Wilhelm Busch	Max und Moritz

Wilhelm Busch	Max und Moritz (neu)

<author>Wilhelm Busch</author><title>Max und Moritz</title>
<author>Wilhelm Busch</author><title>Max und Moritz (neu)</title>

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <record>
    <author>Wilhelm Busch</author>
    <title>Max und Moritz (neu) </title>
  </record>
</book>
```

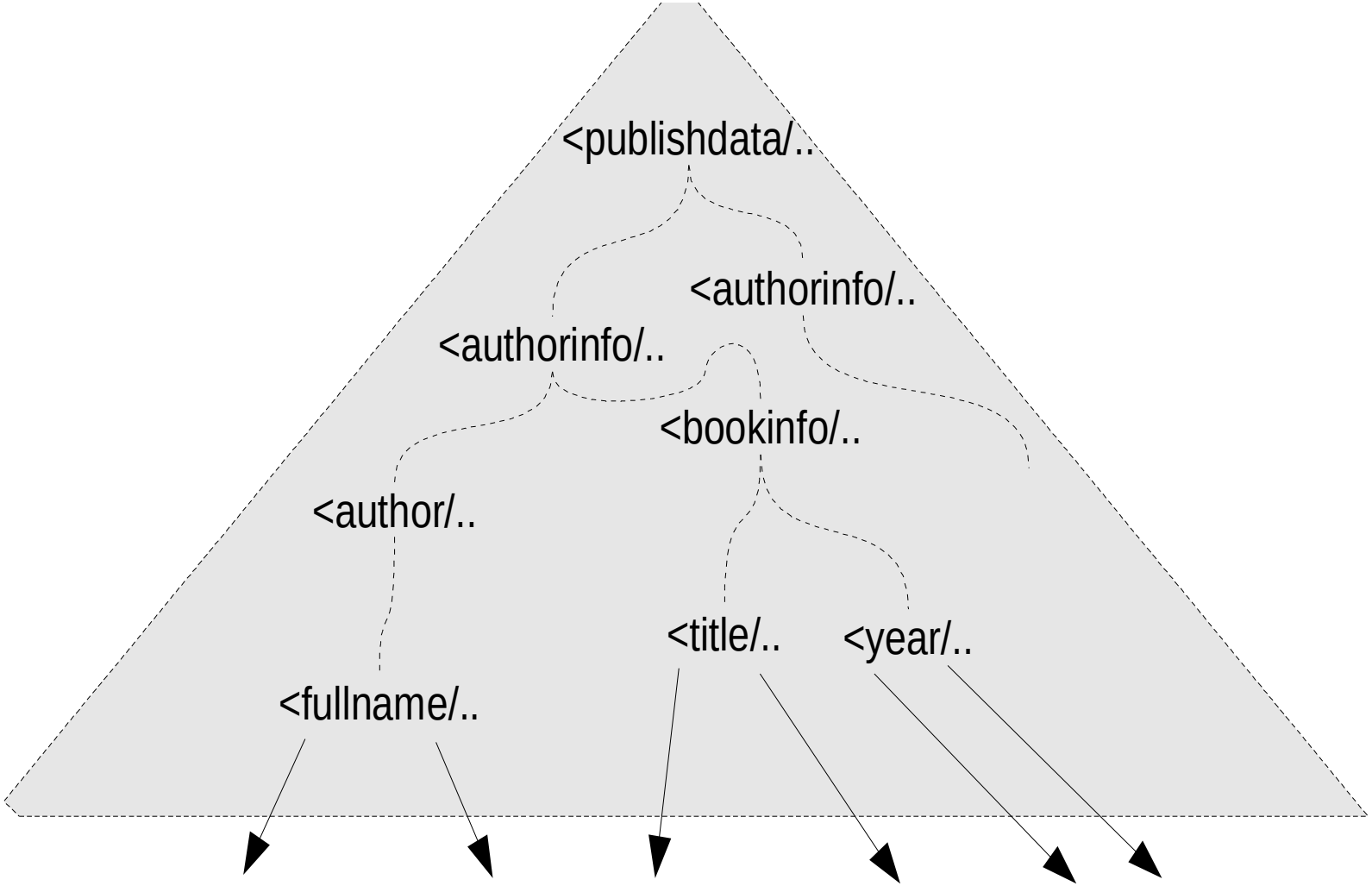
```
<?xml ... ?>
<!DOCTYPE ...>
<book>
  <record>
    <author/>

    <title/>
  </record>
</book>
```

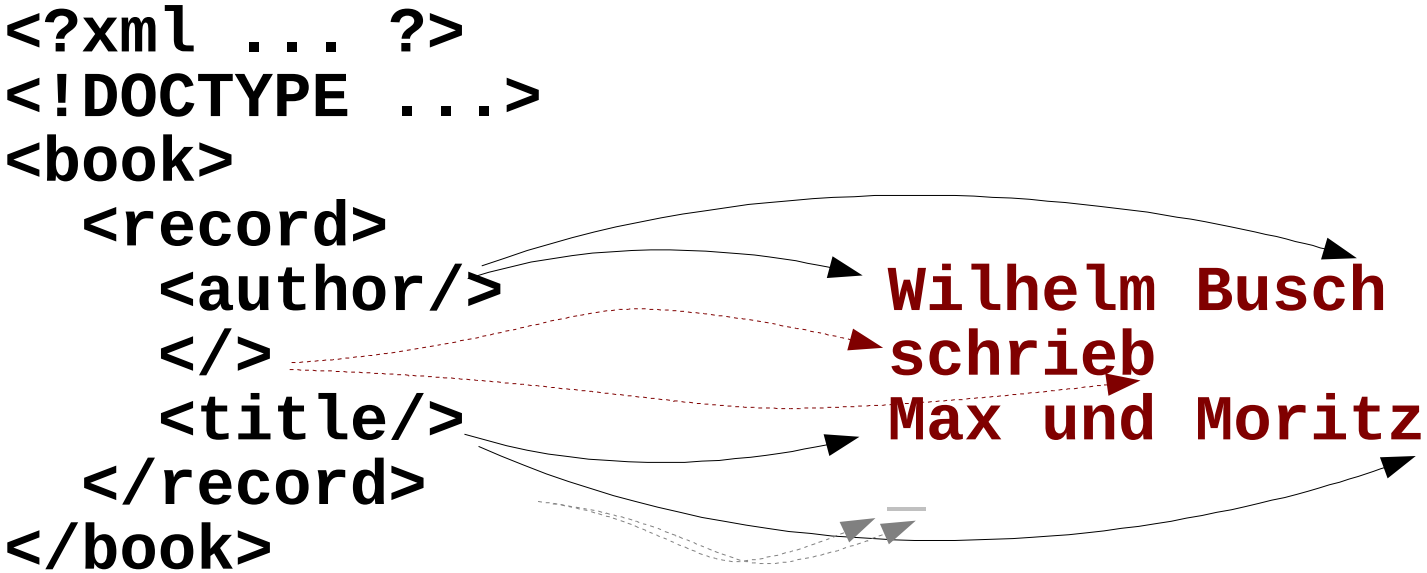
Wilhelm Busch
schrieb
Max und Moritz

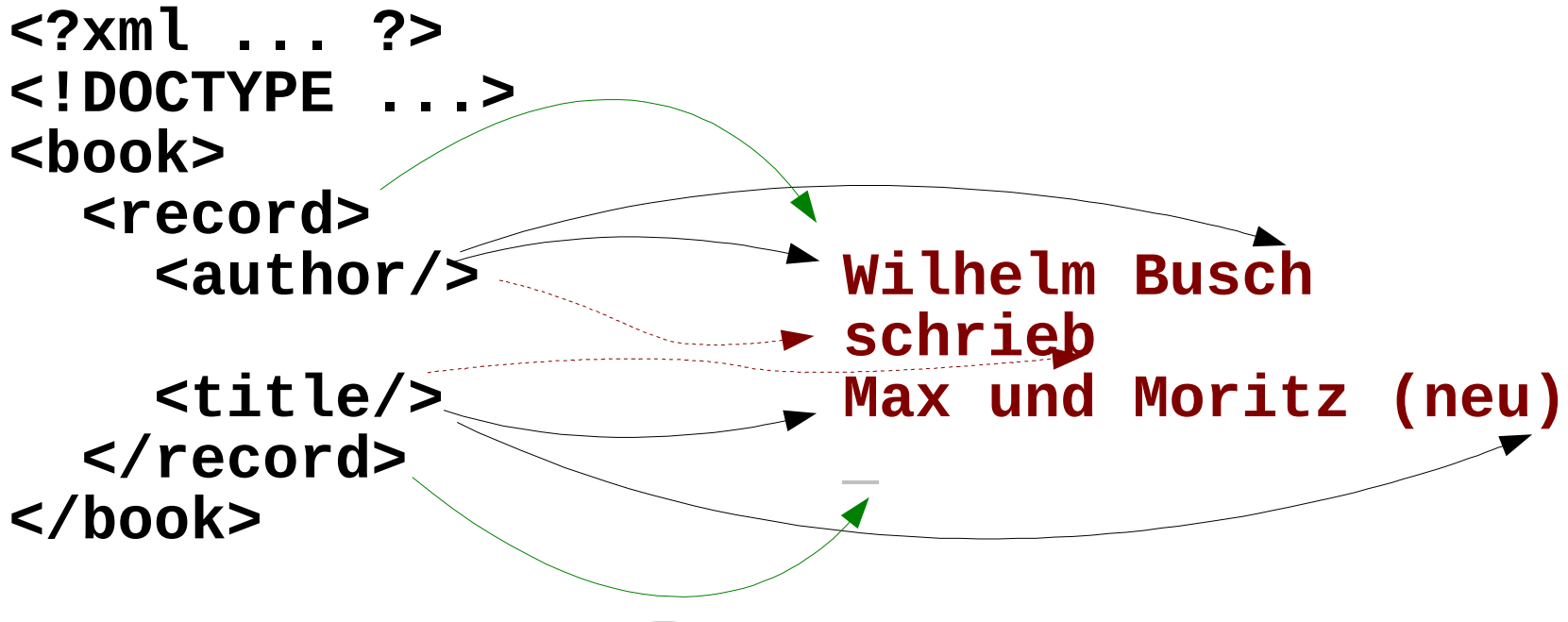
```
<?xml ... ?>
<!DOCTYPE ...>
<book>
  <record>
    <author/>

    <title/>
  </record>
</book>
```

Wilhelm Busch
schrieb
Max und Moritz (neu)

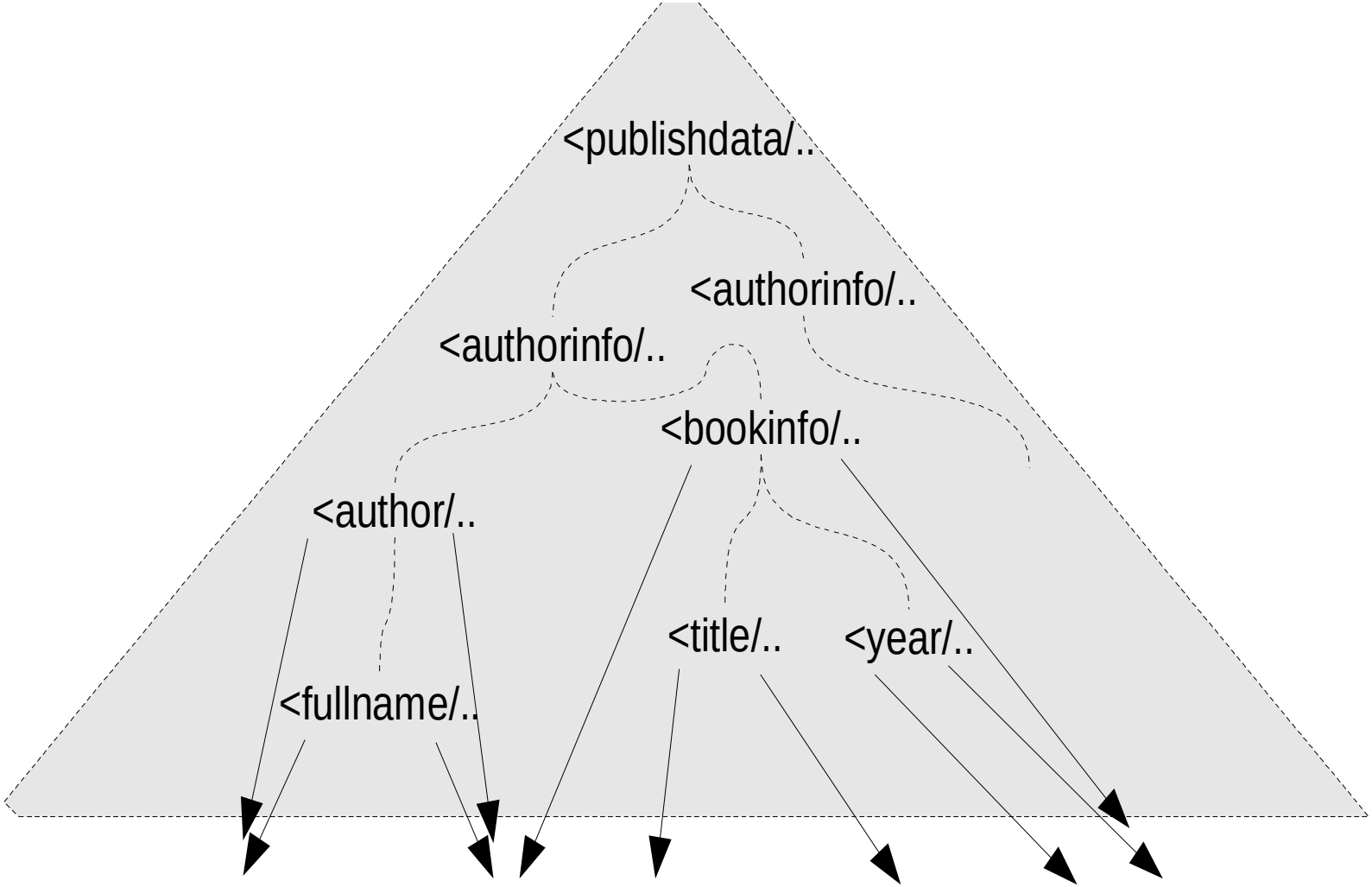
AST/TA – Baum


<publishdata/..
<authorinfo/..
<authorinfo/..
<bookinfo/..
<author/..
<title/..
<year/..
<fullname/..
Wilhelm Busch schrieb Max und Moritz im Jahre 1865

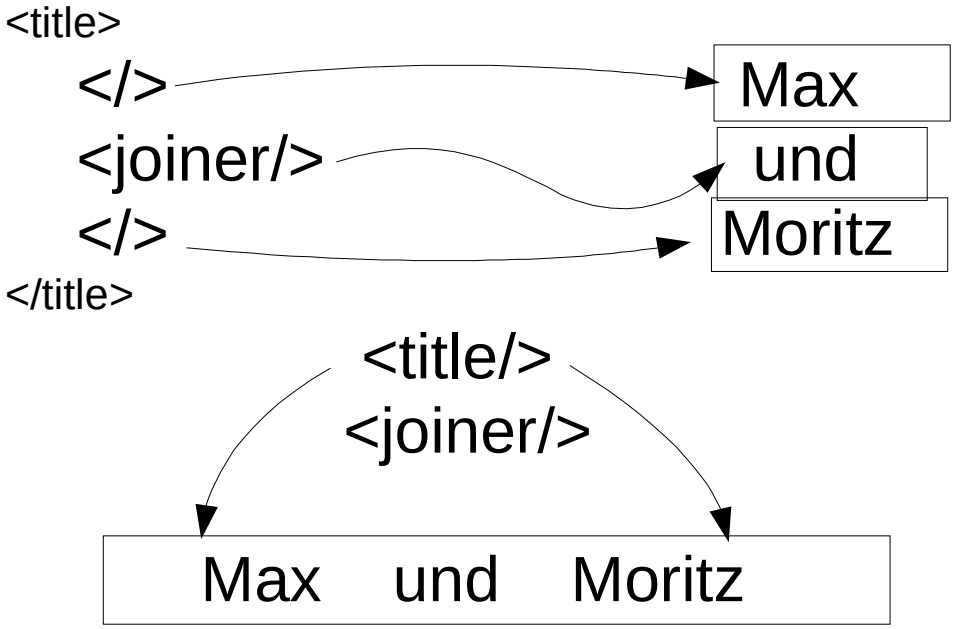

<?xml ... ?>
<!DOCTYPE ...>
<book>
<record>
<author/>
</>
<title/>
</record>
</book>
Wilhelm Busch
schrieb
Max und Moritz


<?xml ... ?>
<!DOCTYPE ...>
<book>
<record>
<author/>
<title/>
</record>
</book>
Wilhelm Busch
schrieb
Max und Moritz (neu)

AST/TA – abschnitte



<title>Max <joiner>und</joiner> Moritz</title>



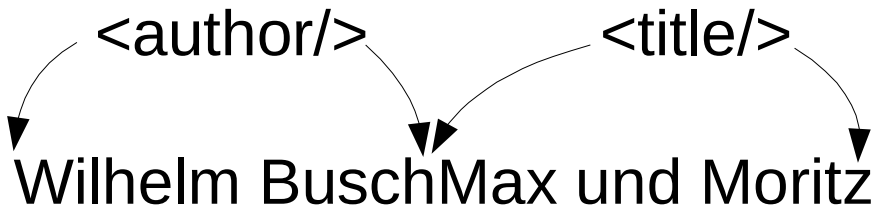
suche nach "Max und Moritz"

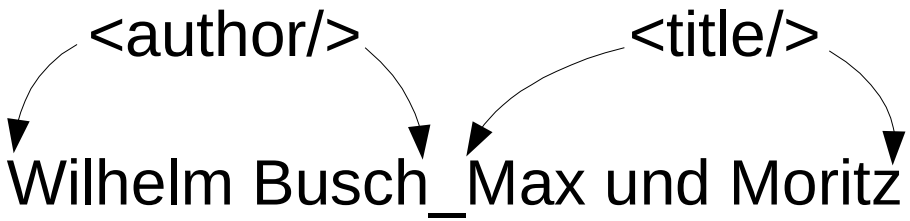
<word><big>X</big>ML</word>

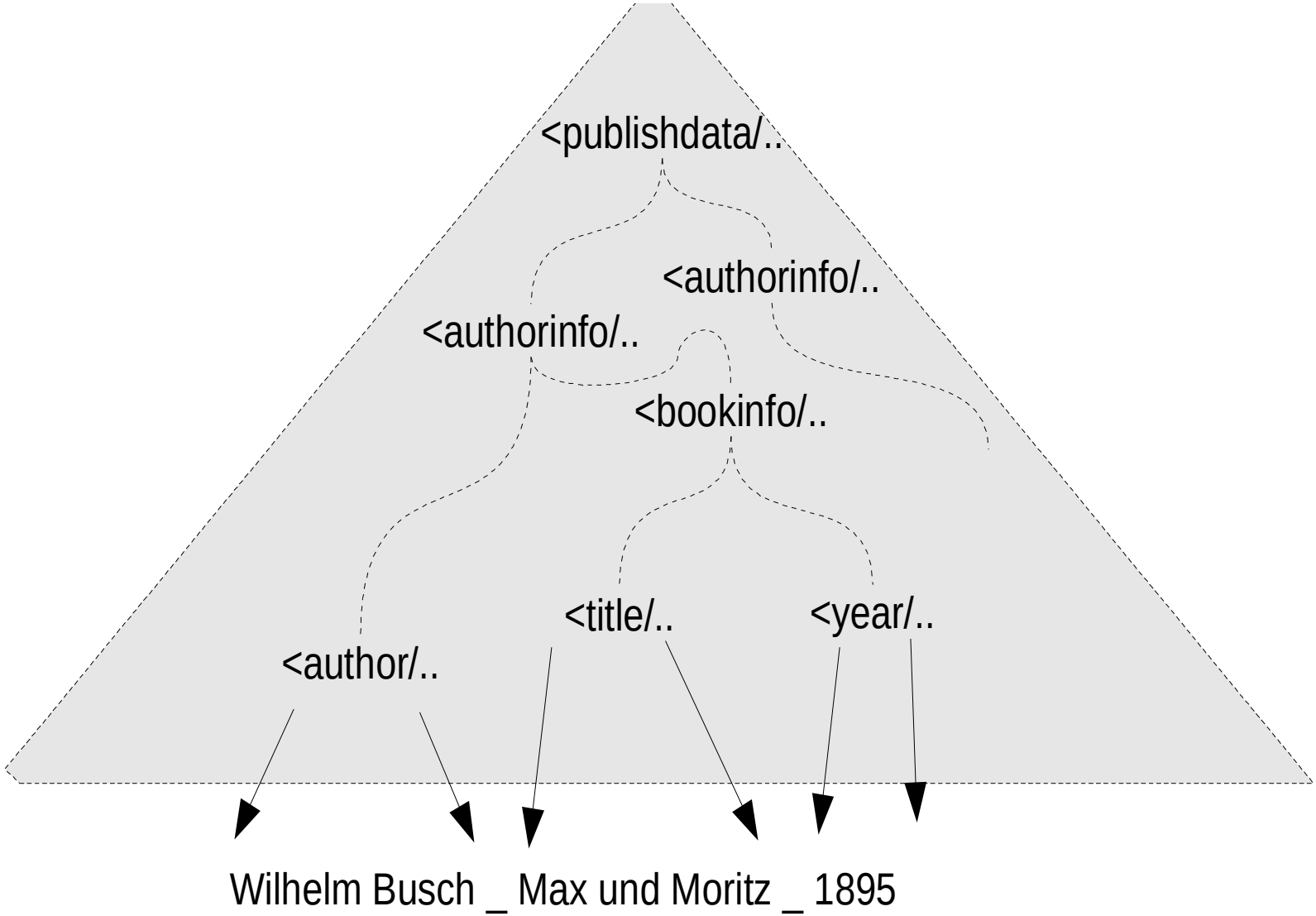
und wo ist XML ?

Wilhelm Busch	Max und Moritz

<author>Wilhelm Busch</author><title>Max und Moritz</title>







... <author>Wilhelm Busch</author><bookinfo ...
><title>Max und Moritz</title><year>1895</year></bookinfo> ...

ein B-Tree weil:

(positional/B/B⁺)*

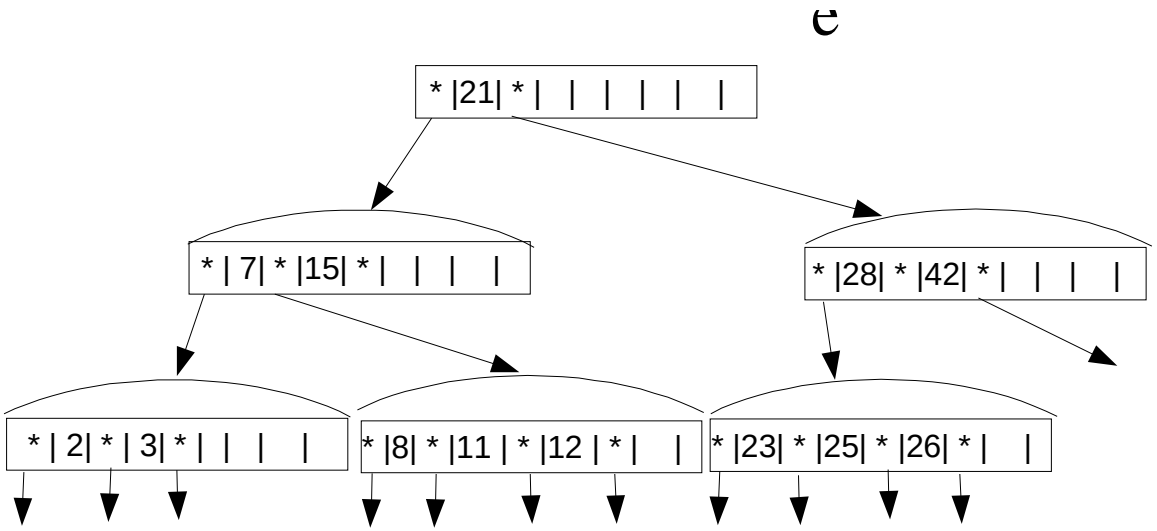
Sekundärspeicher / blockorientiert

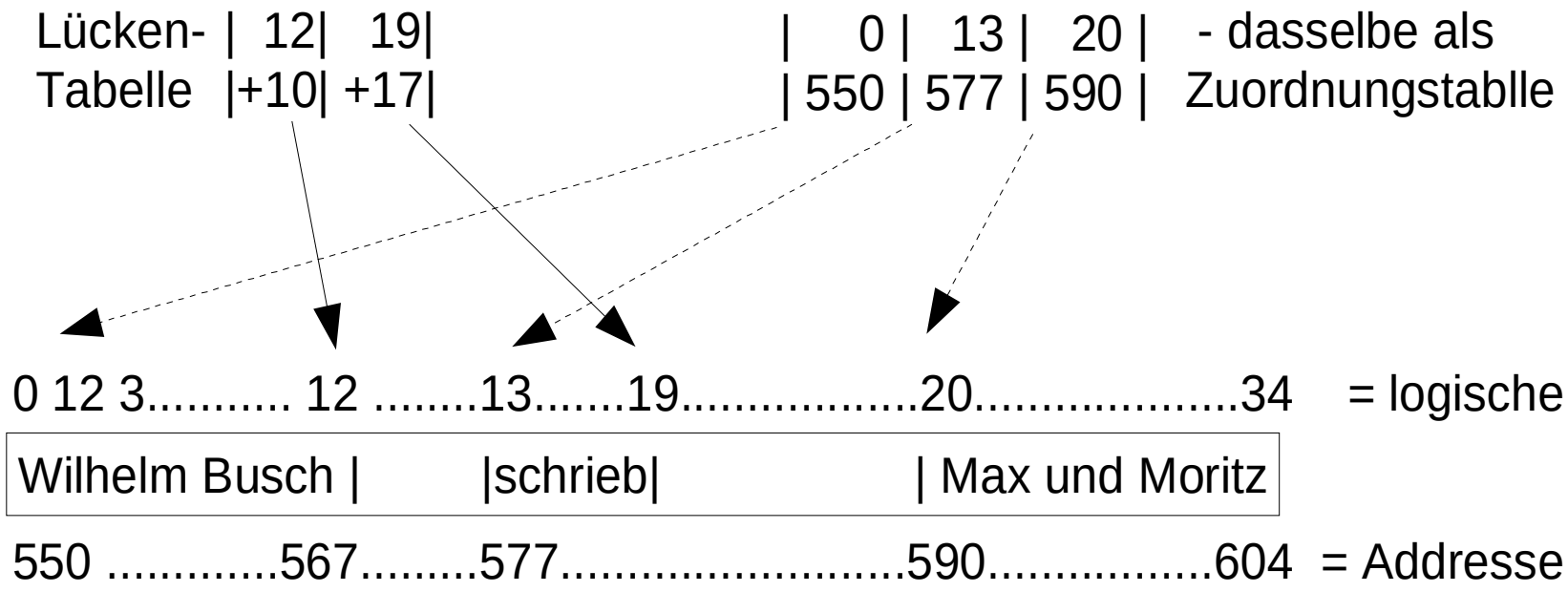
Auffinden / logische Position auf physische — pfad im baum

Weitersetzen / nächste logische auffinden

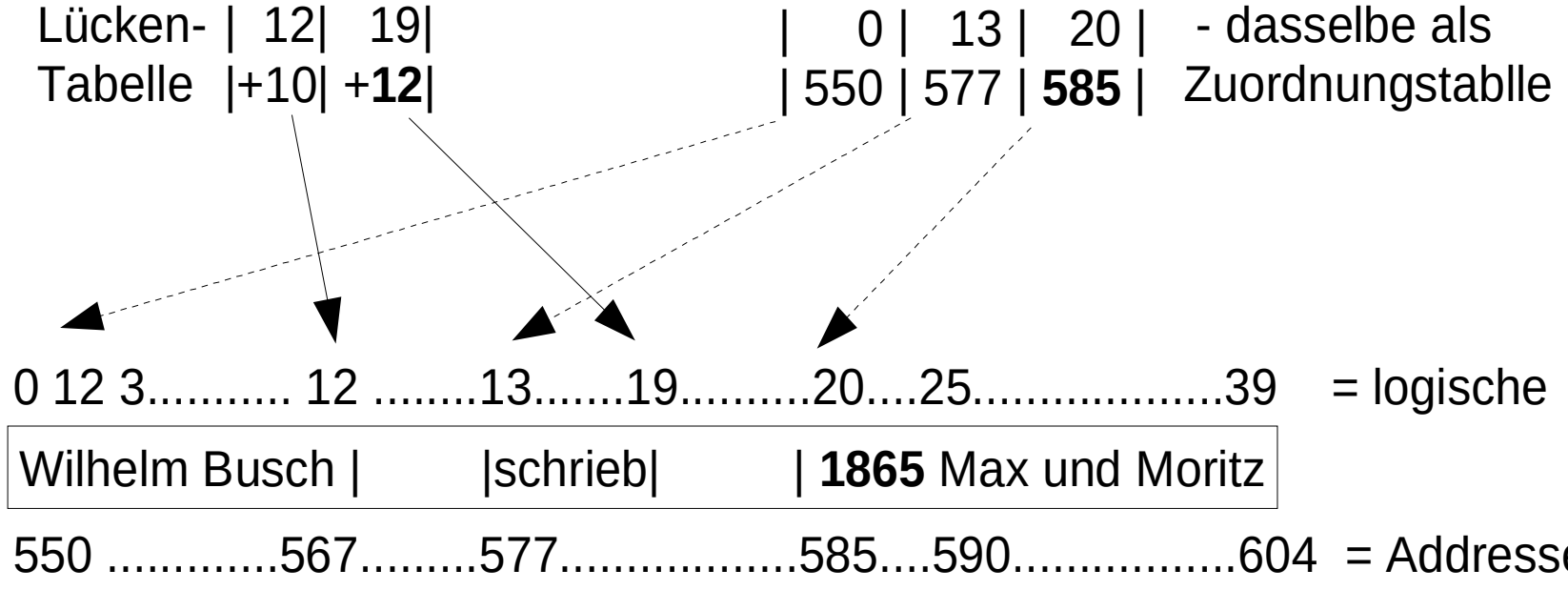
Einfügen/Löschen / Lücken / Umordnung — ausgleichen

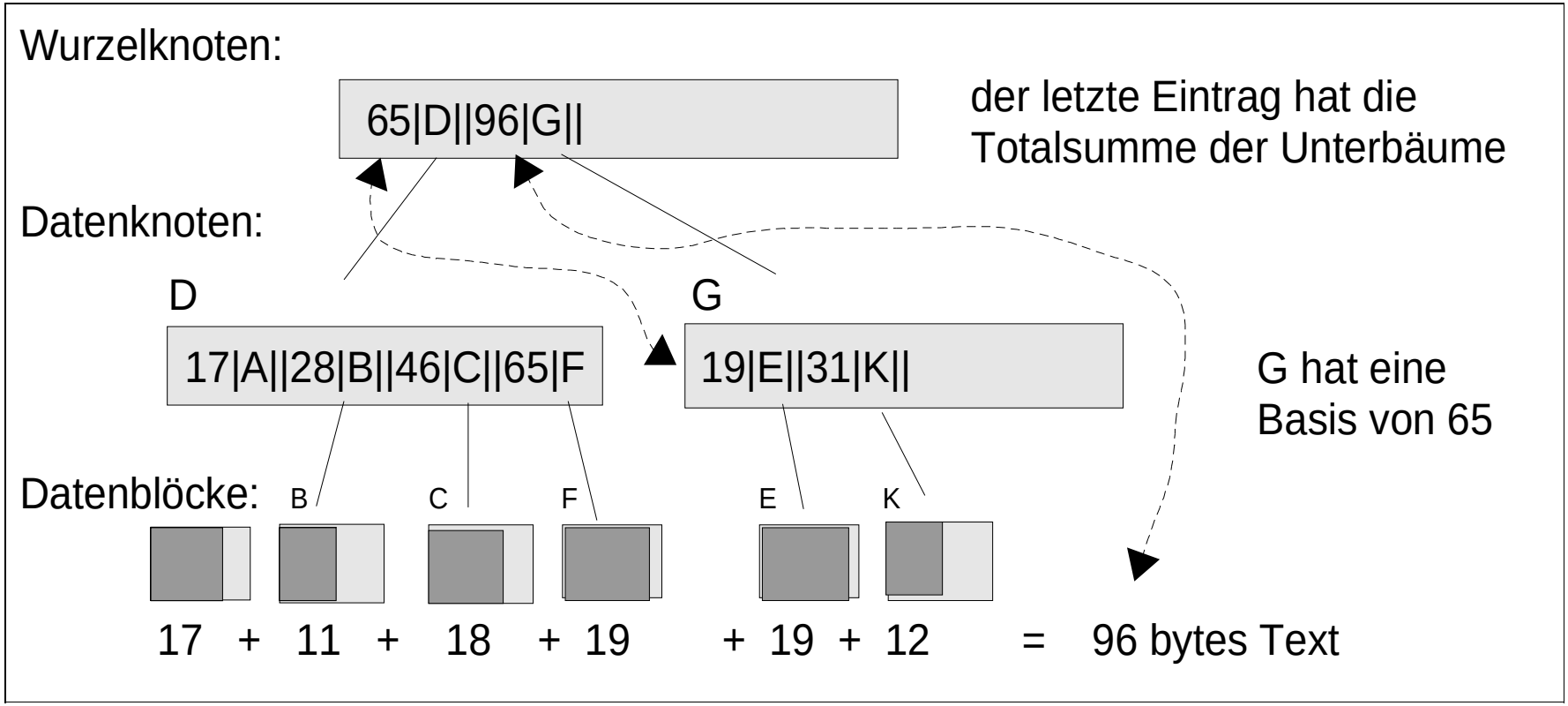
Nutzungsgrad / Füllstände und Lücken





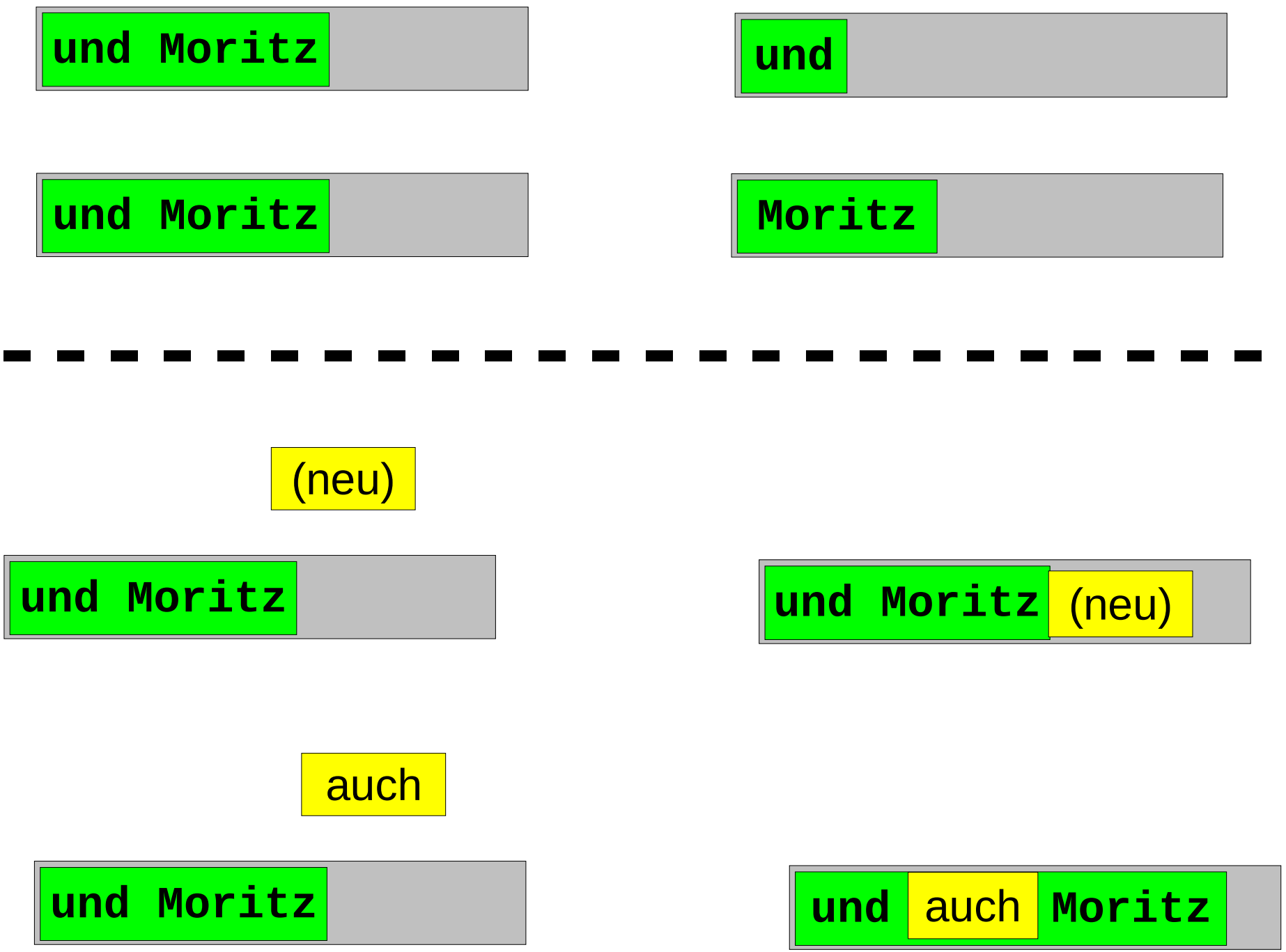
-einfügen der Jahreszahl:

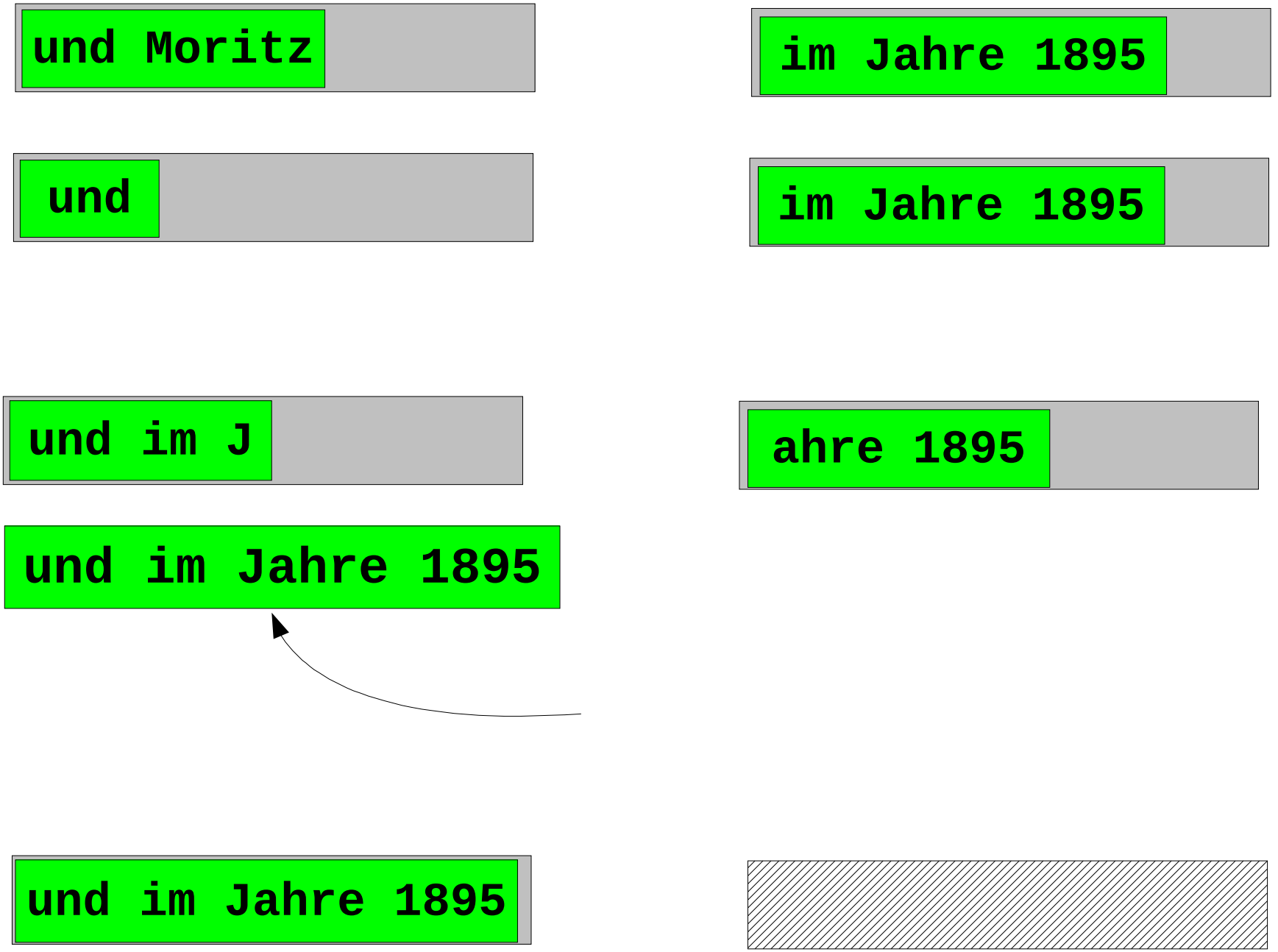


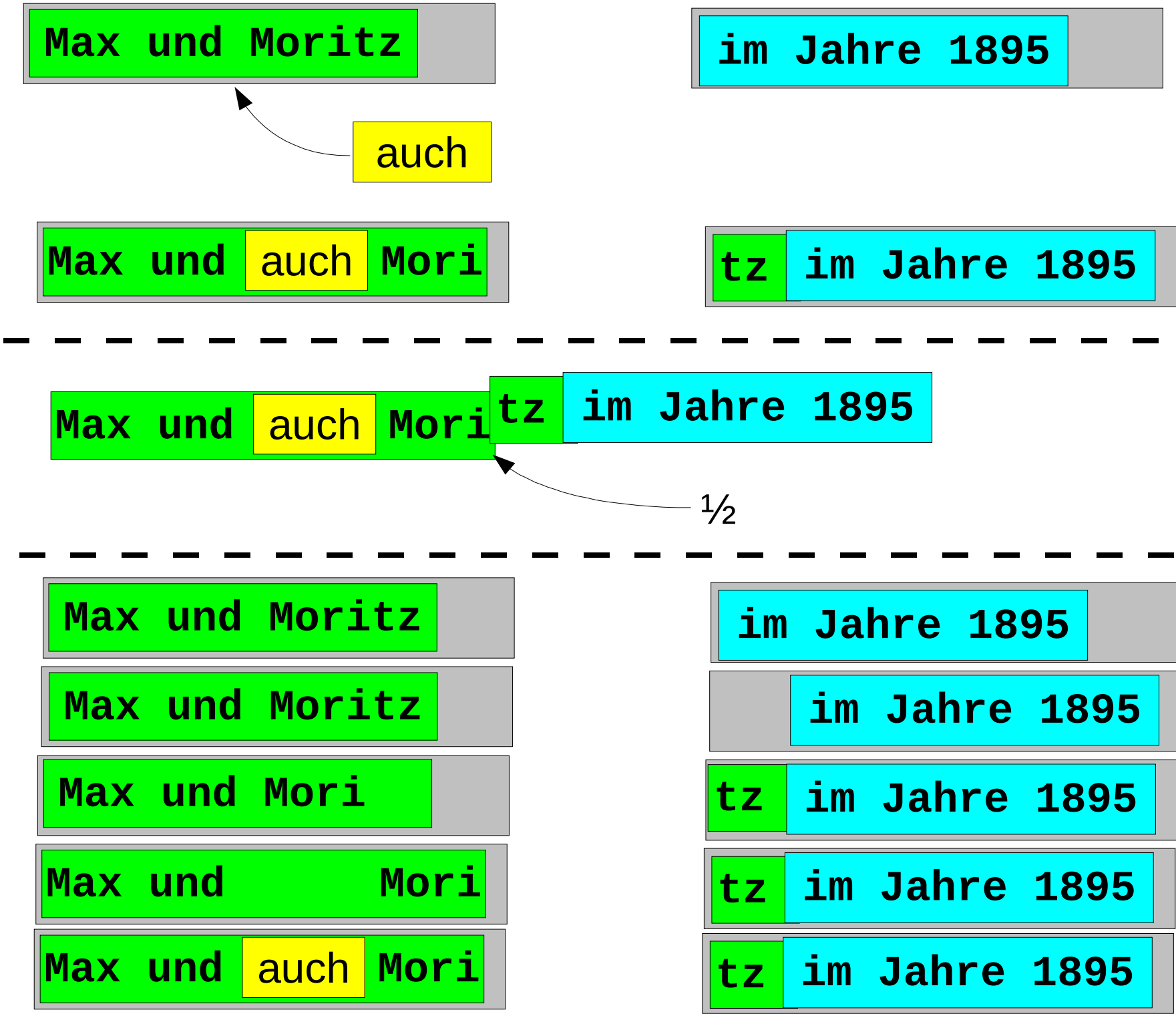


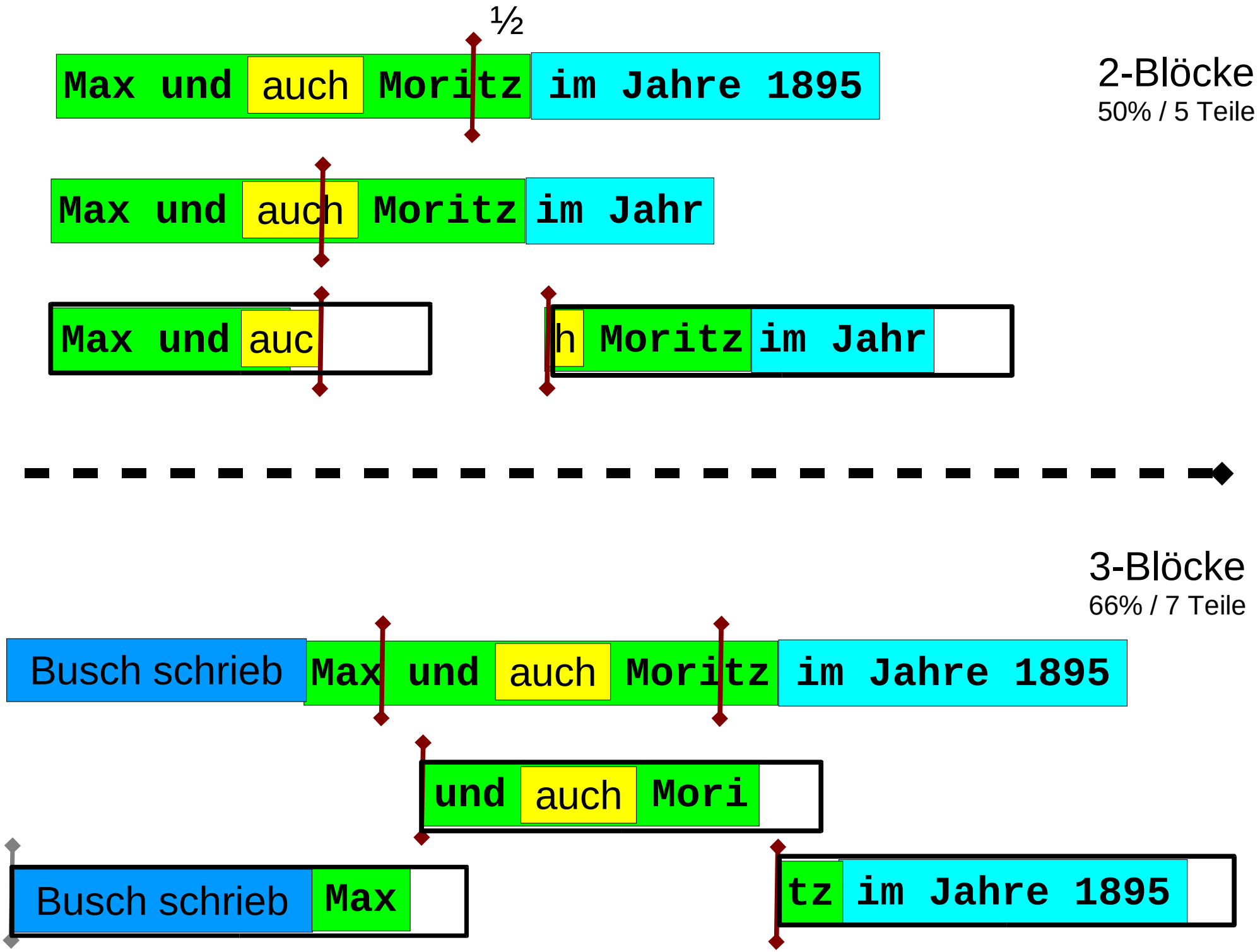
Position 32 = [0=D] /0..46 + [2=C]/28..46 = Rest 4

Auffinden des Pfades


und Moritz
und
und Moritz
Moritz
(neu)
und Moritz
und Moritz (neu)
auch
und Moritz
und auch Moritz


und Moritz
im Jahre 1895
und
im Jahre 1895
und im J
ahre 1895
und im Jahre 1895
und im Jahre 1895


Max und Moritz
im Jahre 1895
auch
Max und auch Mori
tz
im Jahre 1895
Max und auch Mori
tz
im Jahre 1895
½
Max und Moritz
im Jahre 1895
Max und Moritz
im Jahre 1895
Max und Mori
tz
im Jahre 1895
Max und Mori
tz
im Jahre 1895
Max und auch Mori
tz
im Jahre 1895


½
Max und auch Moritz im Jahre 1895
2-Blöcke
50% / 5 Teile
Max und auch Moritz im Jahr
Max und auc
h Moritz im Jahr
3-Blöcke
66% / 7 Teile
Busch schrieb Max und auch Moritz im Jahre 1895
und auch Mori
Busch schrieb Max
tz im Jahre 1895

B[55%] -- C[75%] .... (50%+50%=100%..100%+100%=200%)

B[15%] + C[75%] = 90% = B[90%] + C[---]
B[35%] + C[75%] = 110% = B[55%] + C[55%]

A[75%] + B[75%] -- C[75%] .... (66%+66%+66%=200%...300%)

A[85%] + B[20%] + C[85%] = 190% = A[95%] + B[95%] + C[---]
A[75%] + B[40%] + C[75%] = 210% = A[70%] + B[70%] + C[70%]

bei 31 Einträgen je Zuordnungstabelle

mind.50% = 16 mind.66% = 21

3^{er} Tiefe = $32*32*32 = 32^3 = 32k*512 = 16MB$ $2^{31}/2^9 = 2^{22}$

$\log_{16}(2^{22})=5{,}500$ $\log_{21}(2^{22})=5{,}008$

X.0 Unterschied zwischen B- und B*-Bäumen

B*-Bäume untescheiden sich B-Bäumen in der Definition nur in Punkt 2, dem Füllgrad.
W#hrend bei B-Bäumen jeder Knoten zwischen 50% und 100% gefüllt sein muss (k..2k Einträge) haben muss,
müssen die Knoten bei B*-Bäumen mindestens zu 2/3 aufgefüllt. sein

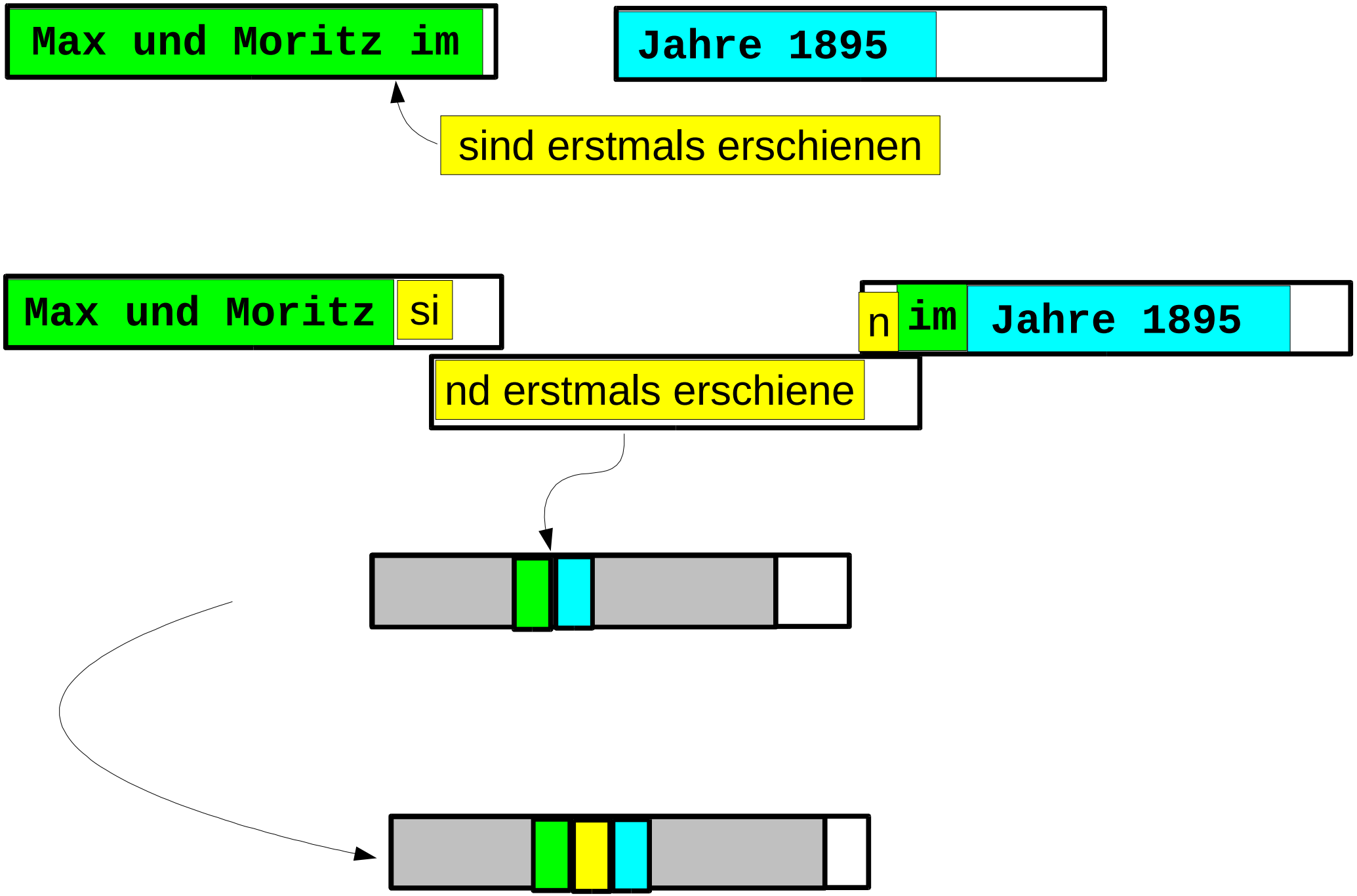
X.x Vor- und Nachteile von B*-Bäumen

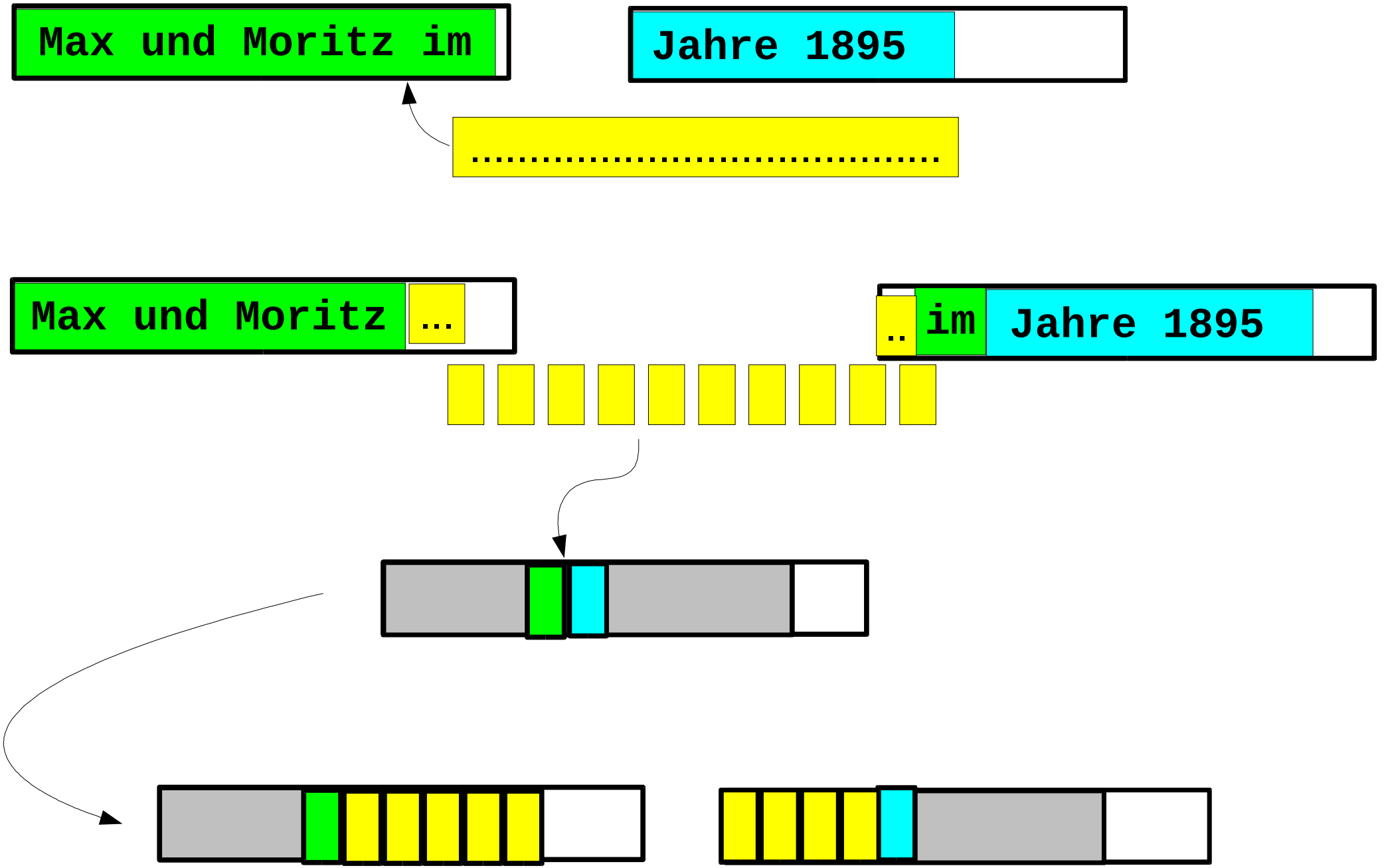
Wenn Knoten minimal besetzt sind, haben Knoten B*-bäumen mehr Söhne als Knoten von B-Bäumen.
Wenn all Knoten minimal besetzt sind ("worst-case"), führt das natürlich dazu, dass die B*-Bäume flacher sind als einfache B-Bäume.
Ausserdem garantieren sie eine bessere Speicherauslastung. Es aber auch Nachteile, wenn man die Knoten immer sehr voll macht.
Denn es kommt mit einer erhöhten Wahrscheinlichkeit dazu, dass ein Knoten geteilt werden muss, da ein Ausgleichen mit
angrenzenden Knoten nicht mehr möglich ist. Und da auch die Vaterknoten recht voll sind, kann sich dies leicht bis zur Wurzel
des Baumes fortsetzen. Während das Auffinden weniger Schritte benötigt, sind Modifikationen aber Baum im Schnitt weniger
loka, abgesehen von den aufwändigeren Operationen.

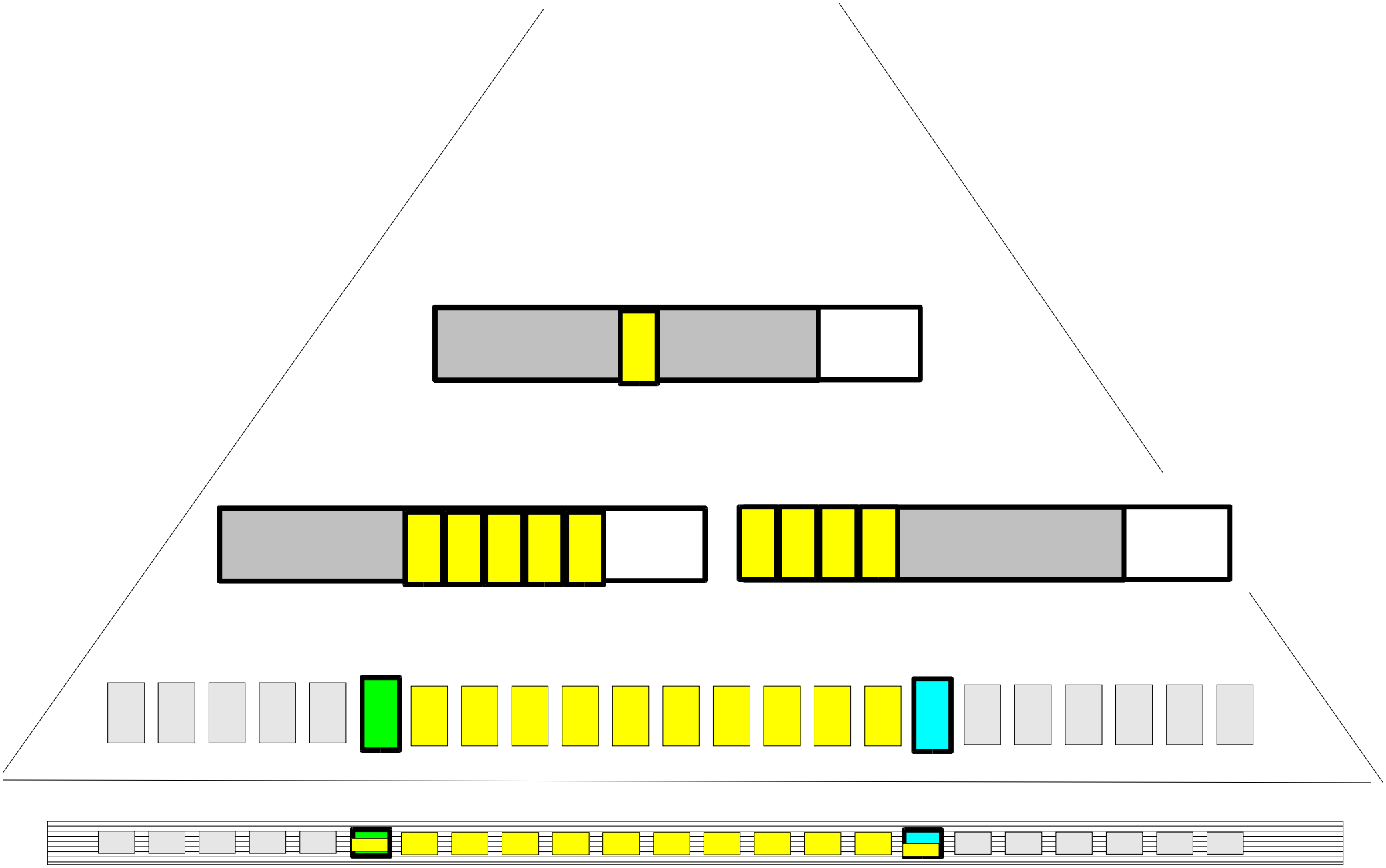
Y.0 B+ Bäume

Bei einem B+Baum unterscheidet sich die Struktur der Endknoten von denen
höherliegender Knoten – nur die Endknoten enthalten Verweise auf Datenblöcke.

B+-Bäume sind (regelmässig) verkettet, zumindest von links-nach-rechts, um das
Weitersetzen zu beschleunigen.


Max und Moritz im
Jahre 1895
sind erstmals erschienen
Max und Moritz
si
n
im
Jahre 1895
nd erstmals erschiene


Max und Moritz im
Jahre 1895
.....
Max und Moritz
…
..
im
Jahre 1895



B[55%] -- C[75%] .... (50%+50%=100%..100%+100%=200%)

B[15%] + C[75%] = 90% = B[90%] + C[---]
B[35%] + C[75%] = 110% = B[55%] + C[55%]

A[75%] + B[75%] -- C[75%] .... (66%+66%+66%=200%...300%)

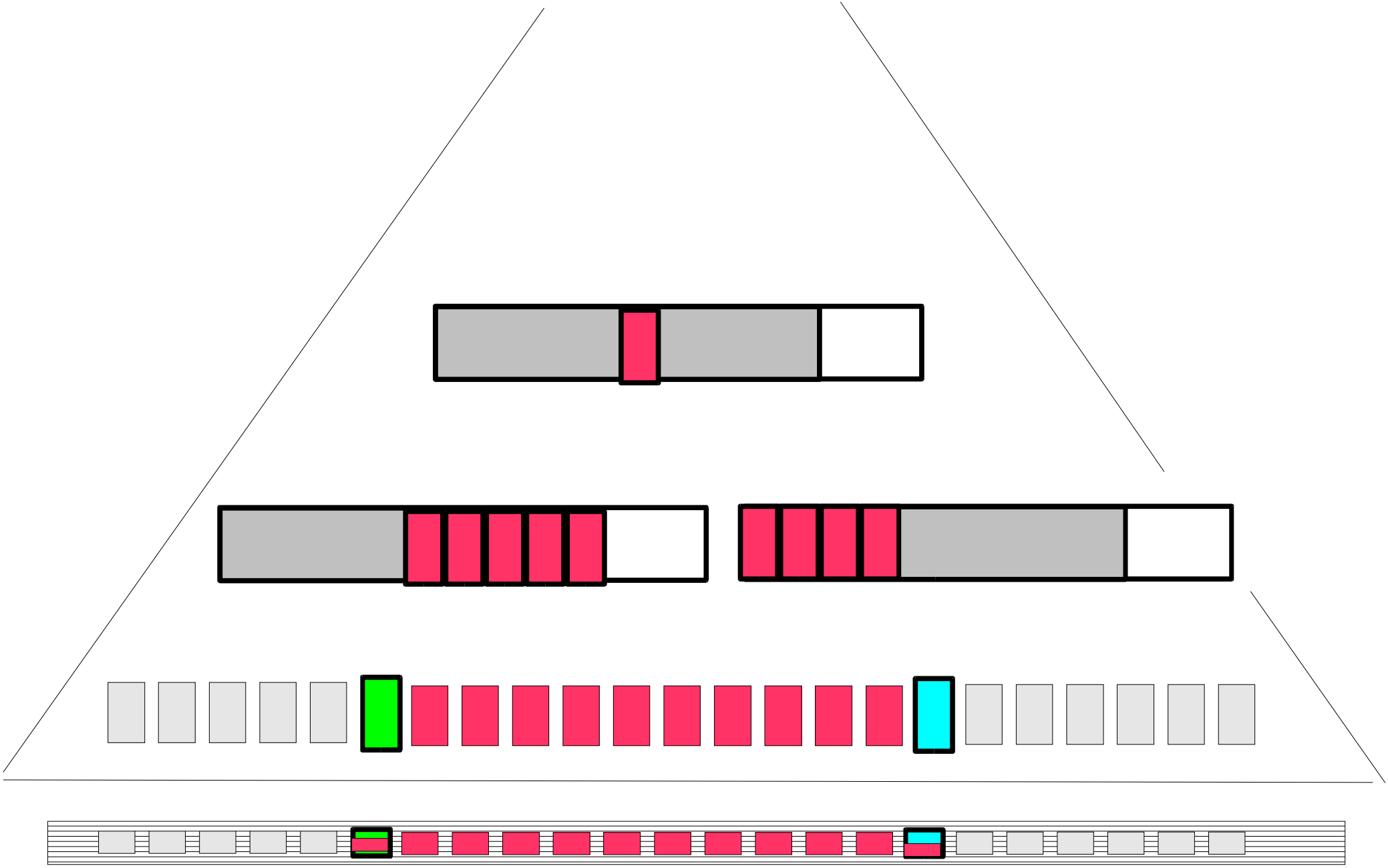
A[85%] + B[20%] + C[85%] = 190% = A[95%] + B[95%] + C[---]
A[75%] + B[40%] + C[75%] = 210% = A[70%] + B[70%] + C[70%]

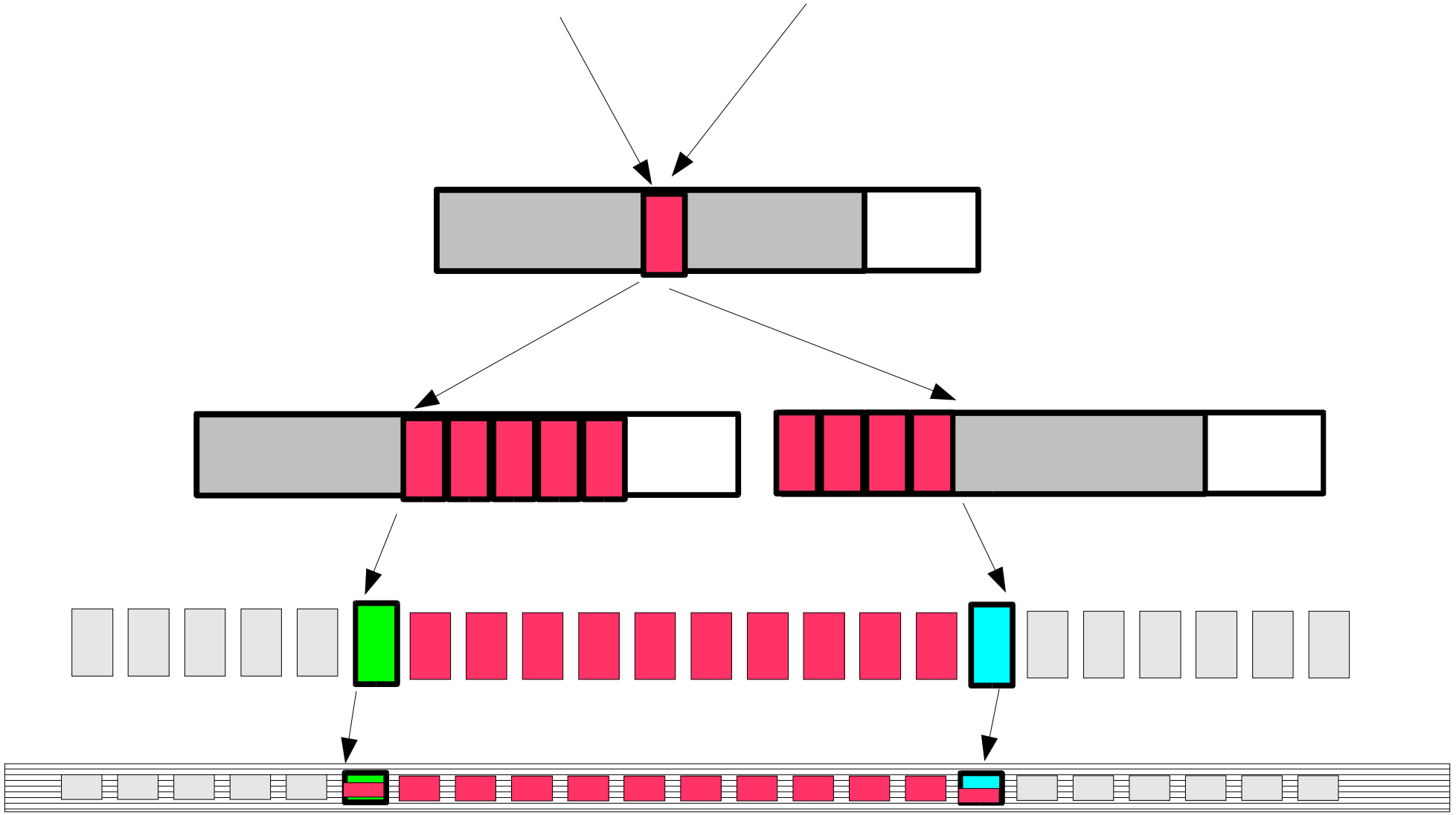
bei 31 Einträgen je Zuordnungstabelle

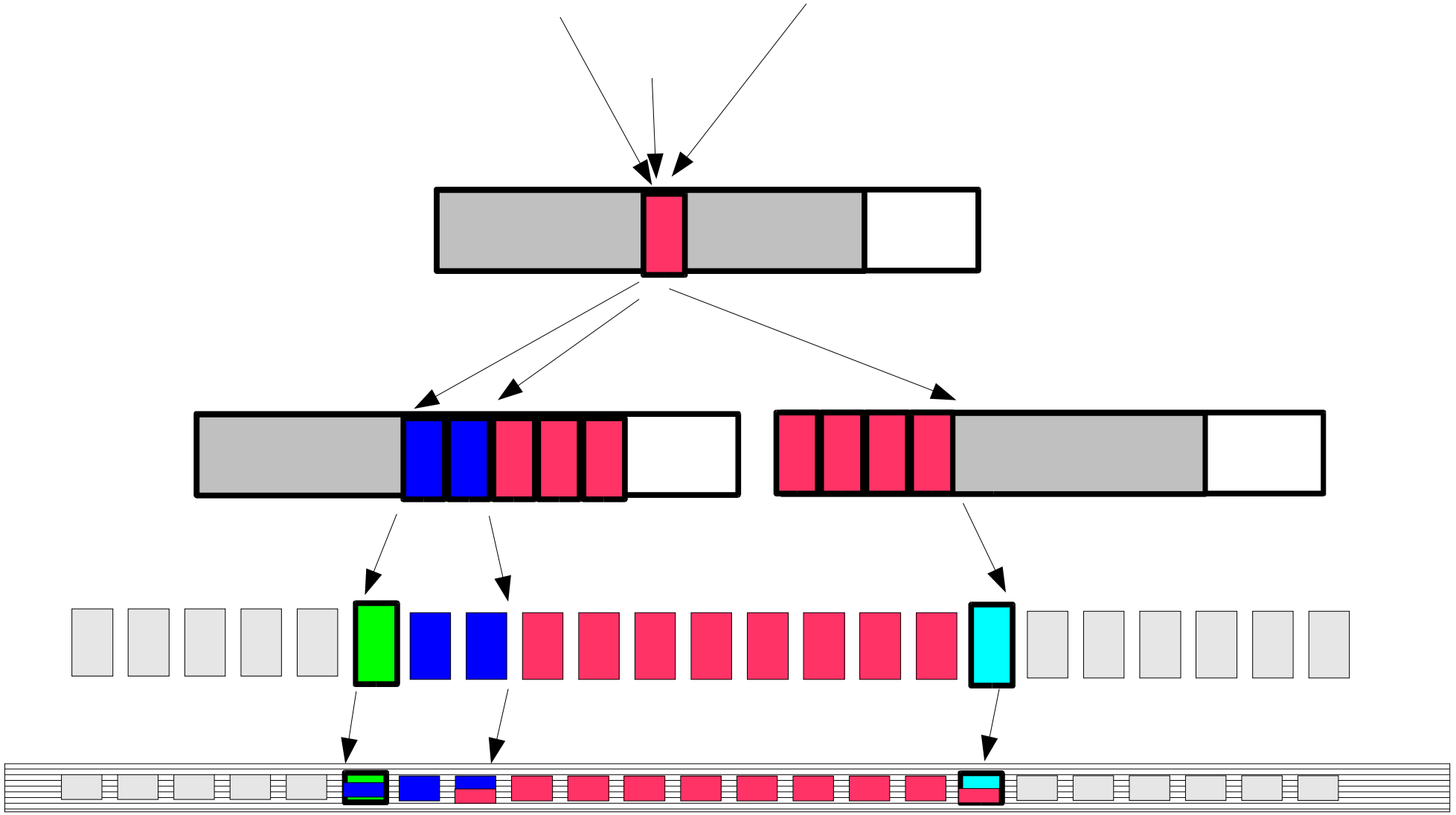
mind.50% = 16 mind.66% = 21

3^{er} Tiefe = $32*32*32 = 32^3 = 32k*512 = 16MB$ $2^{31}/2^9 = 2^{22}$

$\log_{16}(2^{22})=5{,}500$ $\log_{21}(2^{22})=5{,}008$







Wurzelknoten:

65|D||96|G||

der letzte Eintrag hat die Totalsumme der Unterbäume

Datenknoten:

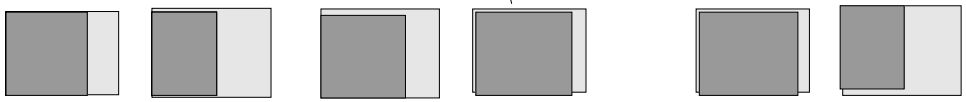
D

17|A||28|B||46|C||65|F

G

19|E||31|K||

G hat eine Basis von 65



17 + 11 + 18 + 19 + 19 + 12 = 96 bytes Text

55|D||86|G||

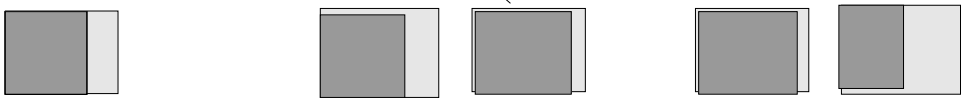
D

17|A||36|C||55|F

G

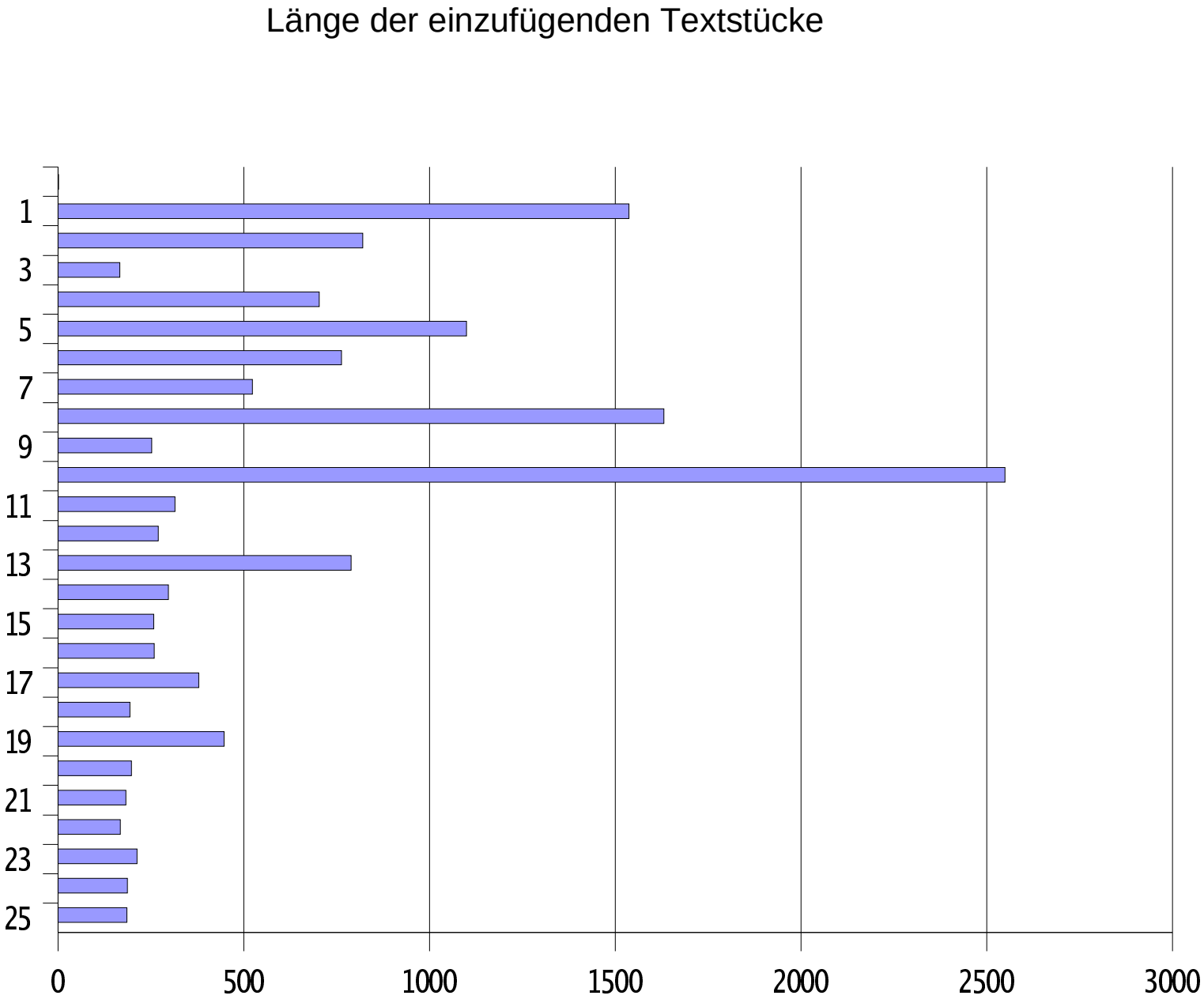
19|E||31|K||

G hat eine Basis von 55

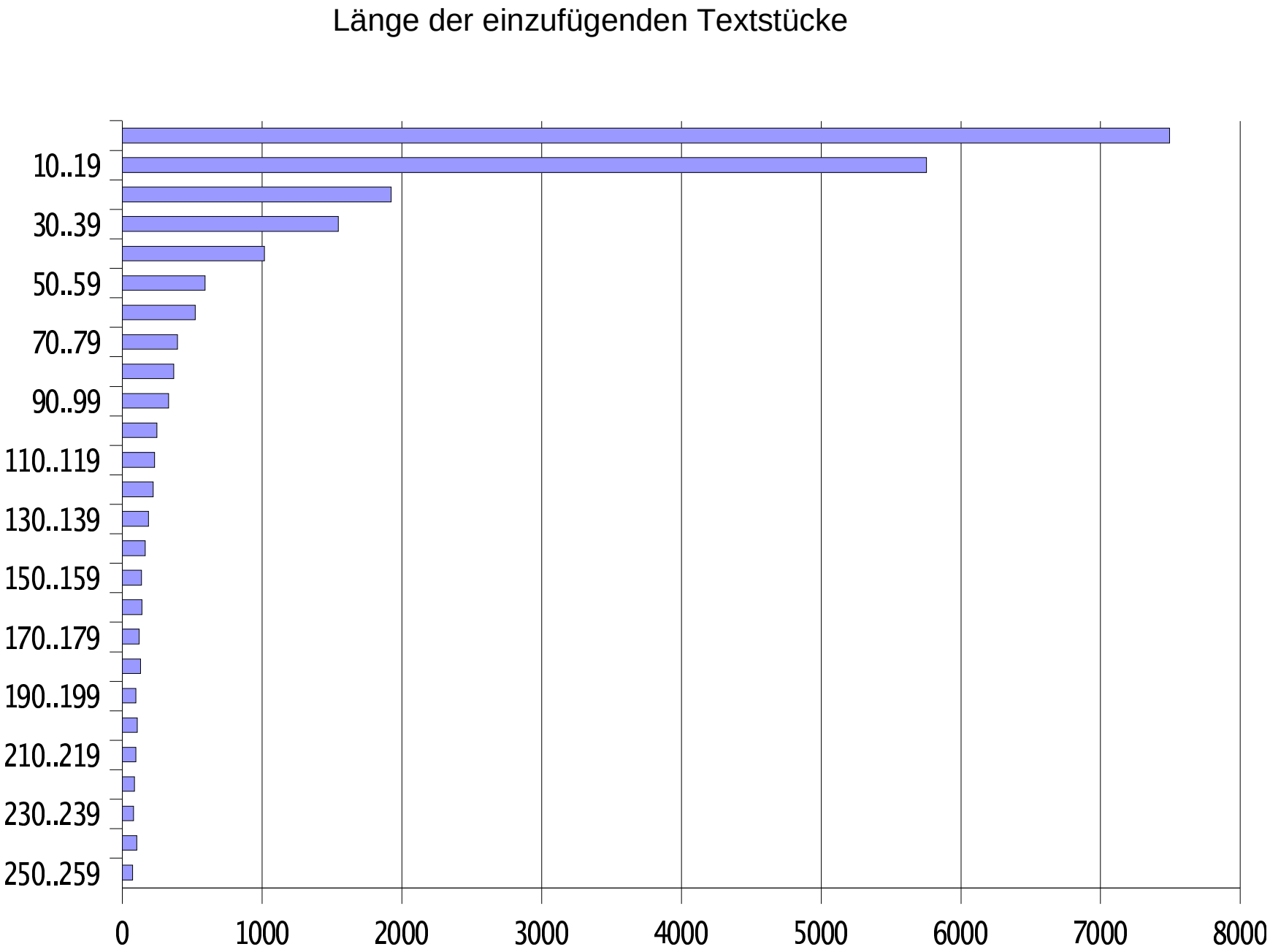


17 + 0 + 19 + 19 + 19 + 12 = 86 bytes Text

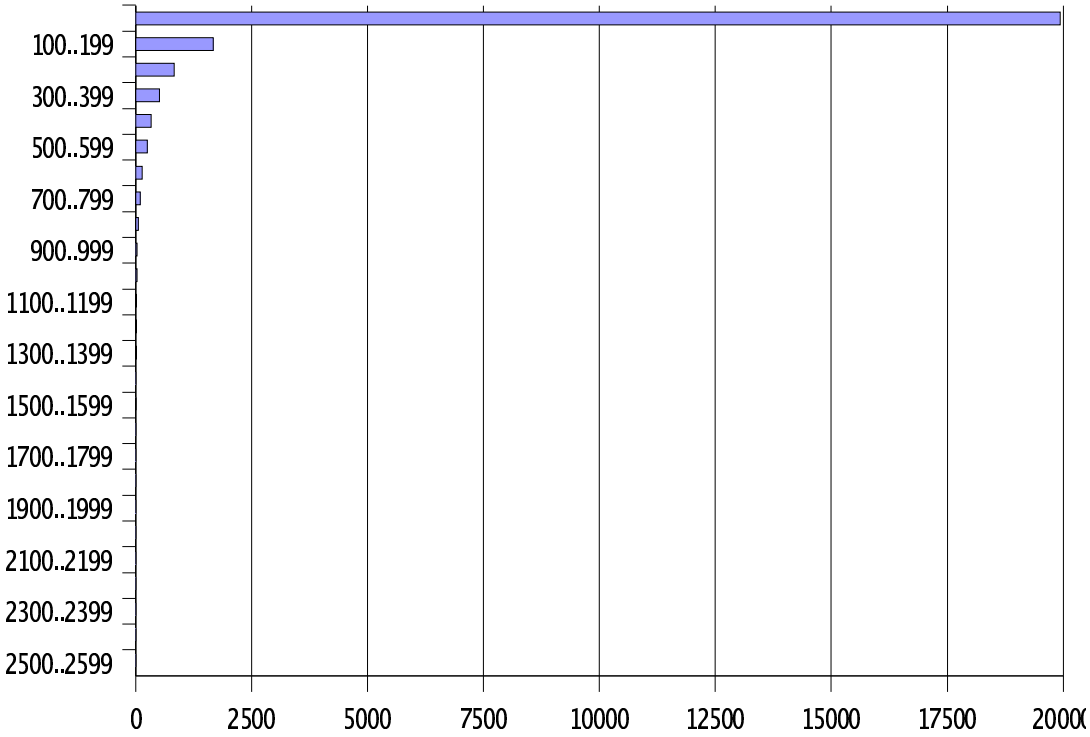
xmlgen 0.02 – ins xee gelesen – insert-calls auf den pTA – 2MB File



xmlgen 0.02 – ins xee gelesen – insert-calls auf den pTA – 2MB File



xmlgen 0.02 – ins xee gelesen – insert-calls auf den pTA – 2MB File

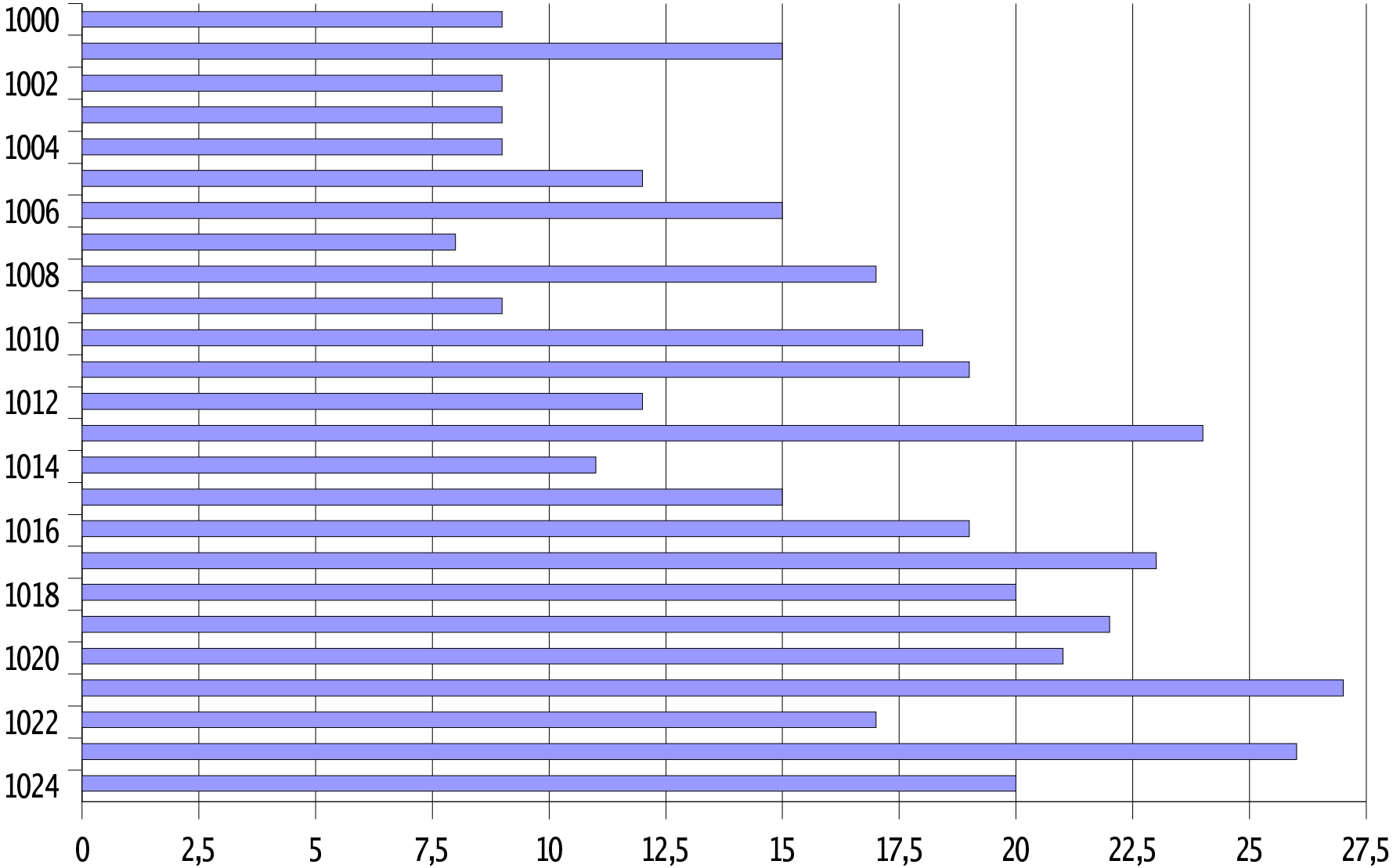


Länge der einzufügenden Textstücke

Einfügelängen	Summe	Prozent
0..125	20560	86,99%
0..250	22096	92,41%
0..500	23285	97,38%
0..1000	23844	99,72%
0..2000	23911	100,00%

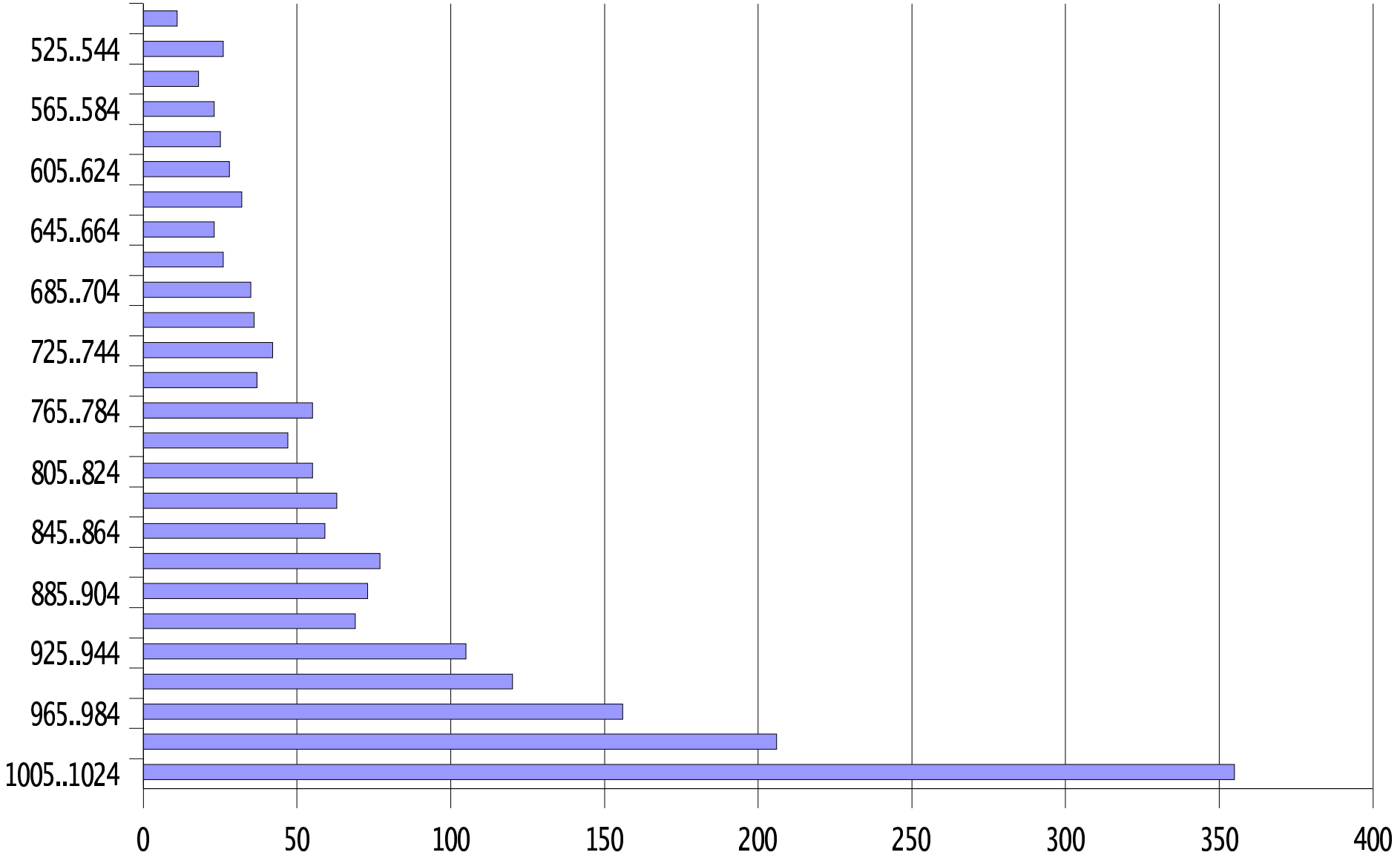
xmlgen 0.02 – ins xee gelesen – verteilung des textesim pTA – 2MB File

Anzahl der Blöcke je *Füllhöhe Blöcke des pTA*



xmlgen 0.02 – ins xee gelesen – verteilung des textesim pTA – 2MB File

Anzahl der Blöcke je *Füllhöhe Blöcke* *des pTA*



---------->

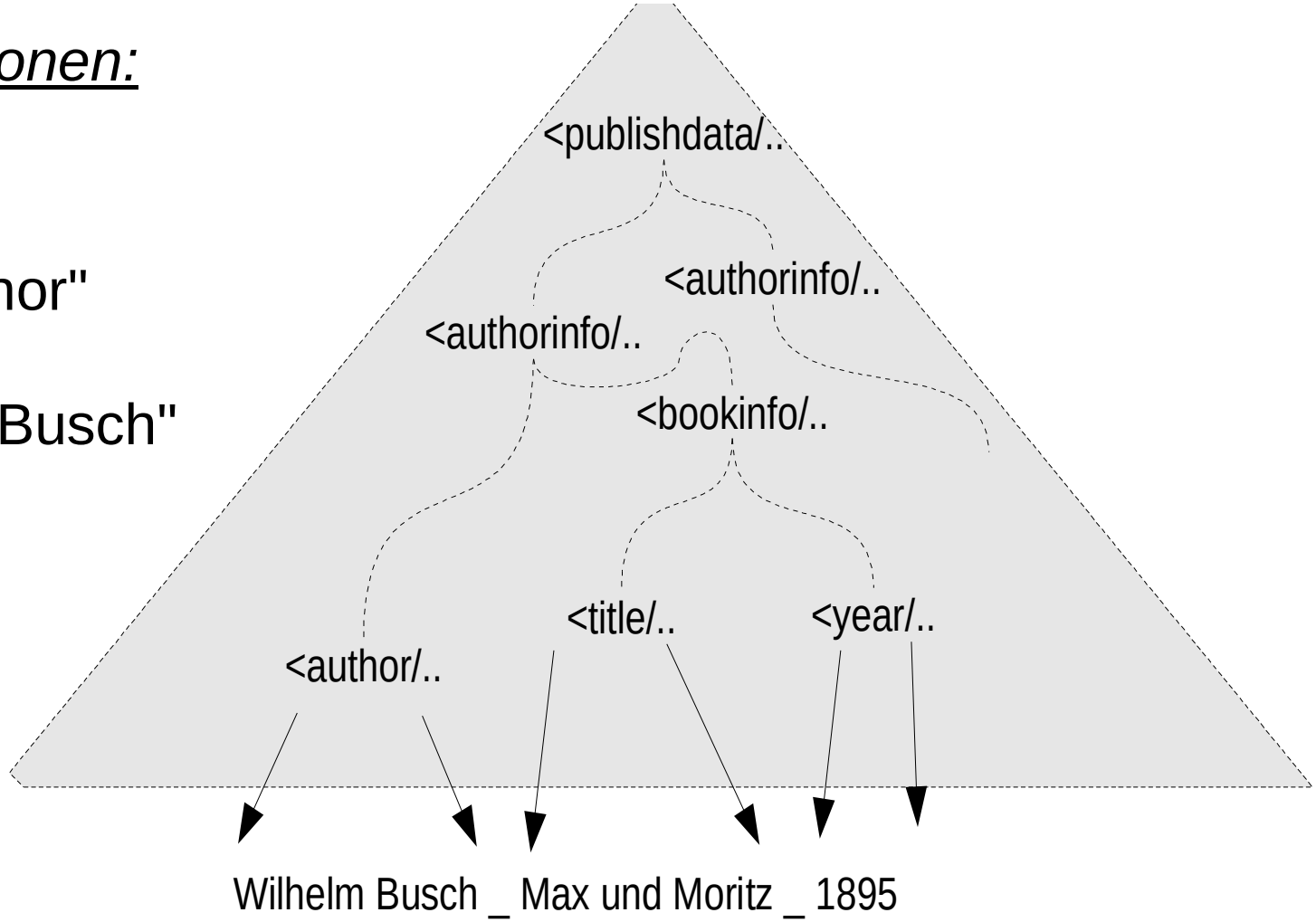
- Angepasste Operationen ..... PCRE
- Vorhaltung Separatorsummen ... statt zählen
- mehrere AST über einem TA ... versch. views
- Synchronisation / Mehrfachzugriff ... locking
- Zugriffsrechte / gemeins. DB ... view differences

- Vorteile AST/TA Modell
- xmlg Erfahrungen (Januar)

spez. Operationen:

xpath: "//author"

PCRE: "/w+\sBusch"

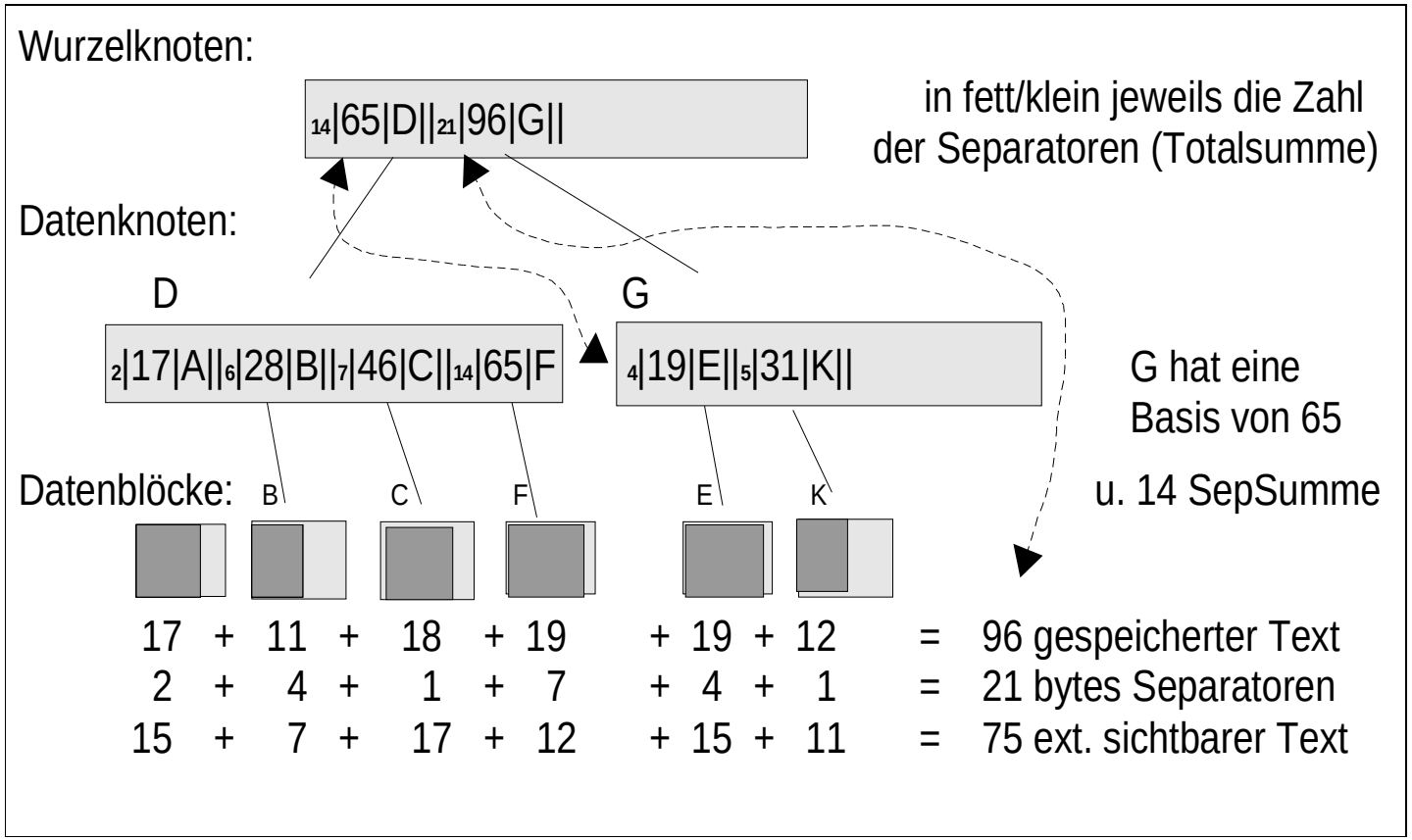


xmlg/libpcre zeigt Verwendung
Anpassung an TextArray auf Sekundärspeicher
Zeichen-Hol-Funktion - vorwärts/backtracking
Optimierung auf möglichst lokalen Zugriff

aus Text Mining / Retrieval
Index-listen der Worte
Thesauri Match und Suche
Abgleichen bei Insert/Delete

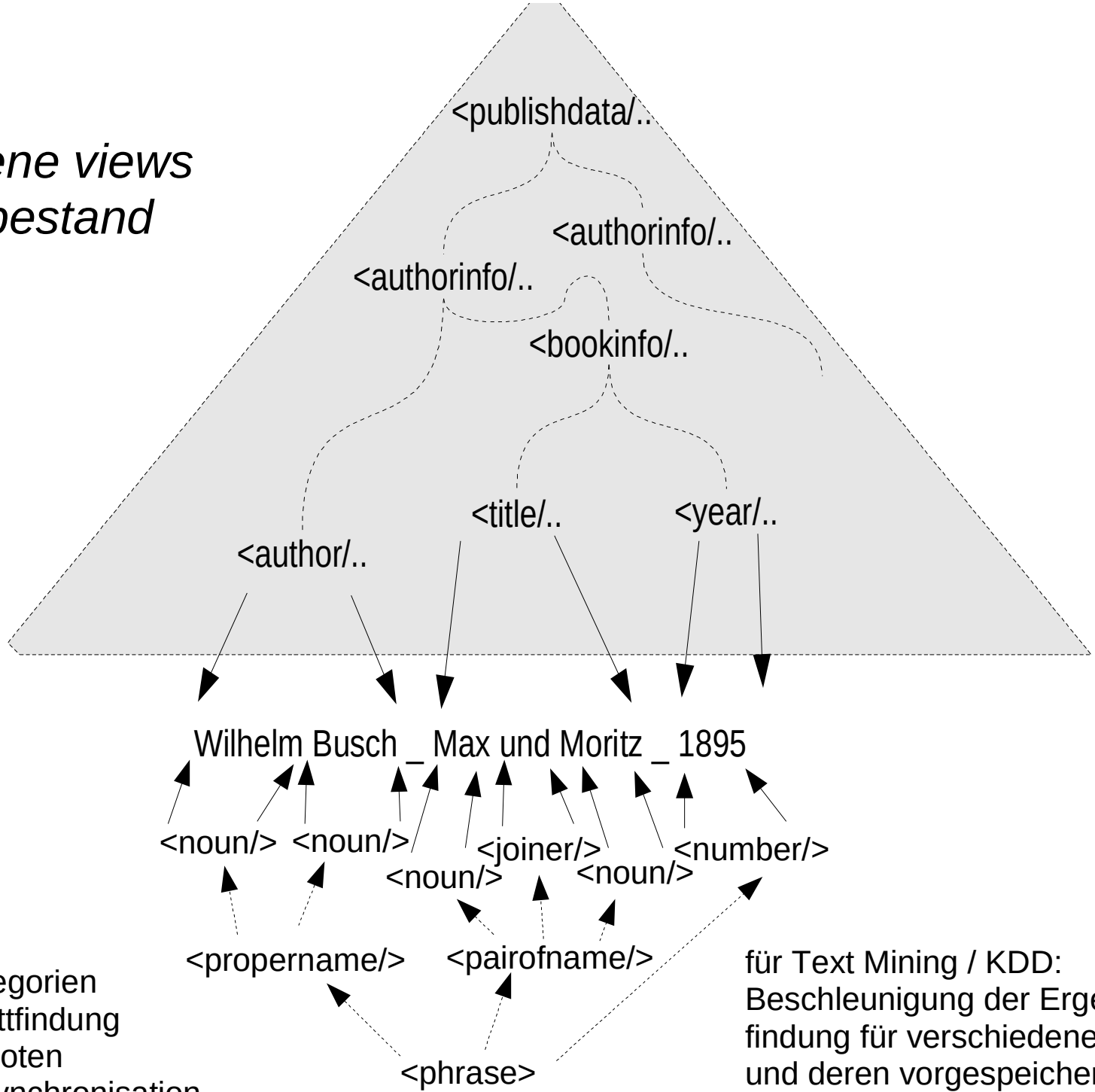
Vorhaltung Separatorsummen:

- externer Gehalt an "visuellem" Text
- intern vorhandene Zeichen im TextArray
- Differenzbildung aus Zuordnungstabellen
  .... statt durchsuchen vor Ausgabe



Multi-AST

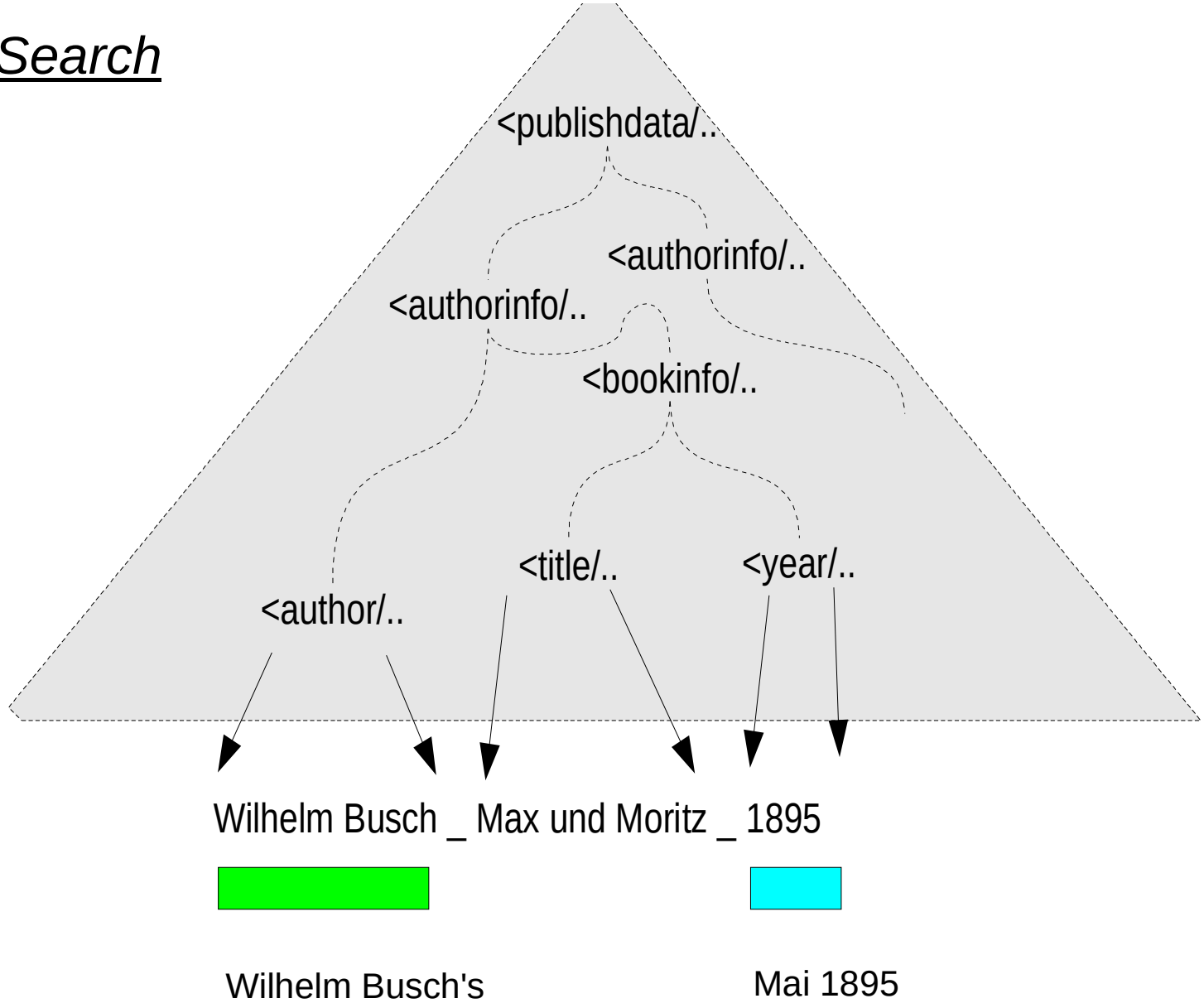
verschiedene views
auf Datenbestand



- überlappende Kategorien
- optimierte abschnittfindung
- attribut-werte in knoten
- aber: TA-update synchronisation

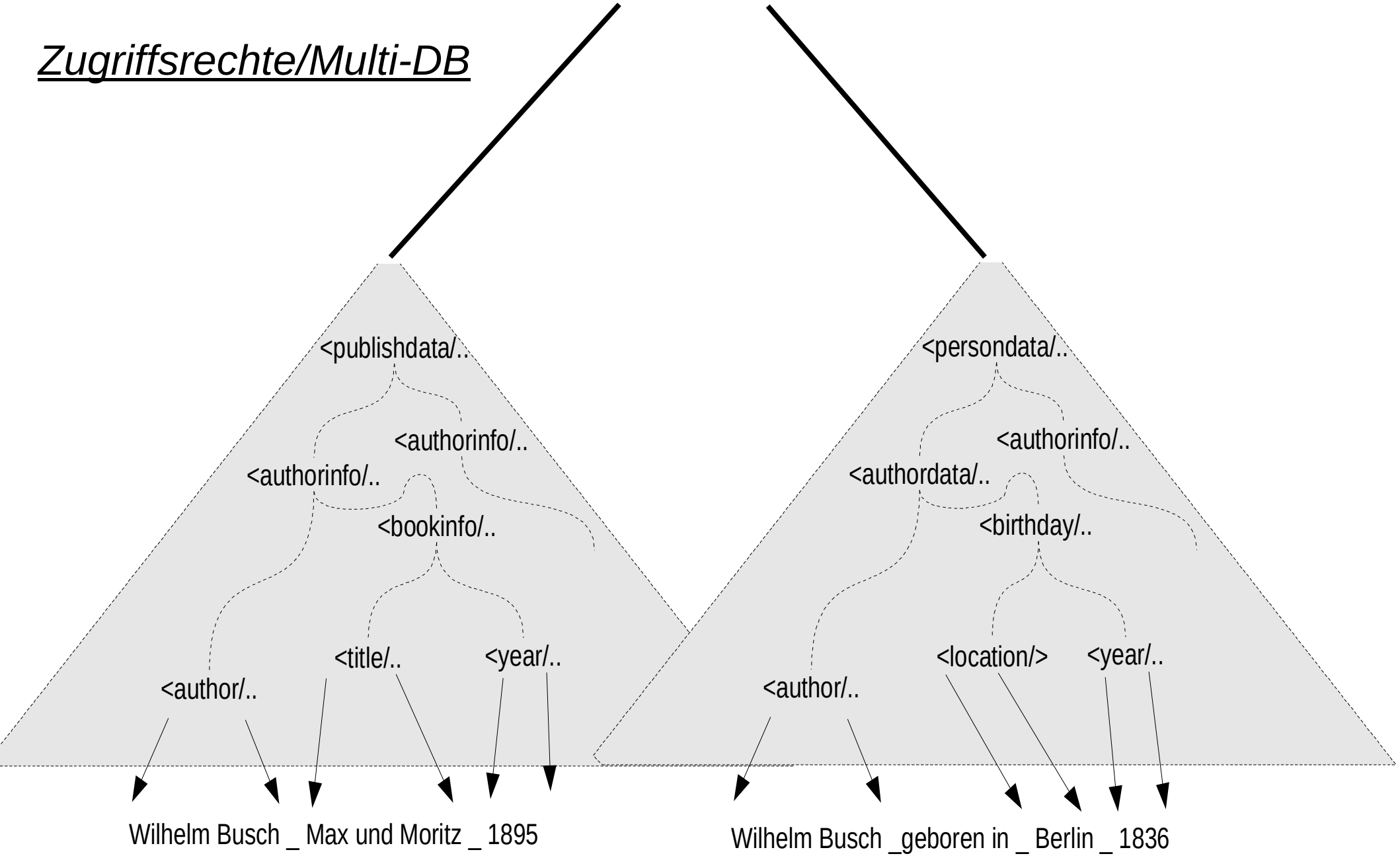
für Text Mining / KDD:
Beschleunigung der Ergebnis findung für verschiedene Anfragen und deren vorgespeicherten Bewertungen und Kategorien

Multi-Update/Search



mgl. Verwendung von Mechanismen der hierarchischen DBMS (sep.hilfe?)

Zugriffsrechte/Multi-DB



RechteZugriffs Mechanismen nach Pfaden

Statistik über Einträge zu "Wilhelm Busch"

Vorteile

- Wiederverwendung bekannter Algorithmen
  effizienter, optimierter, leichter implementierbar u. wartbar u. spezialisierbar
  (wo ist der unterschied von text mining und knowledge discovery in databases (kdd) ?)

- Multi Varianten
  auch bezüglich spezialisierten Varianten

- Export/Import/Transfer der Daten
  XML ist das interne Datenmodell, keine Umrechnung

- und ....

Erfahrung

- xmlg als Hauptspeicher Variante mit libpcre
- Vortrag zu 9. Januar 2003