Physische XML-Speicherstrukturen und Indizes

Holger Meyer

Database Research Group

University of Rostock

E-Mail: hm@GUUG.de

Inhalt

1. Überblick XML-Dateiorganisation und Zugriffpfade
2. Werteindizes
3. Volltextindizes
4. XML-Dateiorganisation
5. Struktur- und Pfadindizes
6. Zusammenfassung

Zugriffspfade

- *Werteindex*, atomar Elementinhalte, Attributwerte, strukturierte Anteile eins XML-Dokumentes
- *Volltextindex* einzelnen Worte des *Volltextes*, ganzen Sätzen oder Phrasen einer natürlichen Sprache, Zugriff mit *Information Retrieval*-Techniken.
- *Struktur- und Pfadindex* effiziente Auswertung von Anfragen unter Einbeziehung der Dokumentstruktur.
- physische *Speicherstrukturen* zur Ablage von XML-Fragmenten

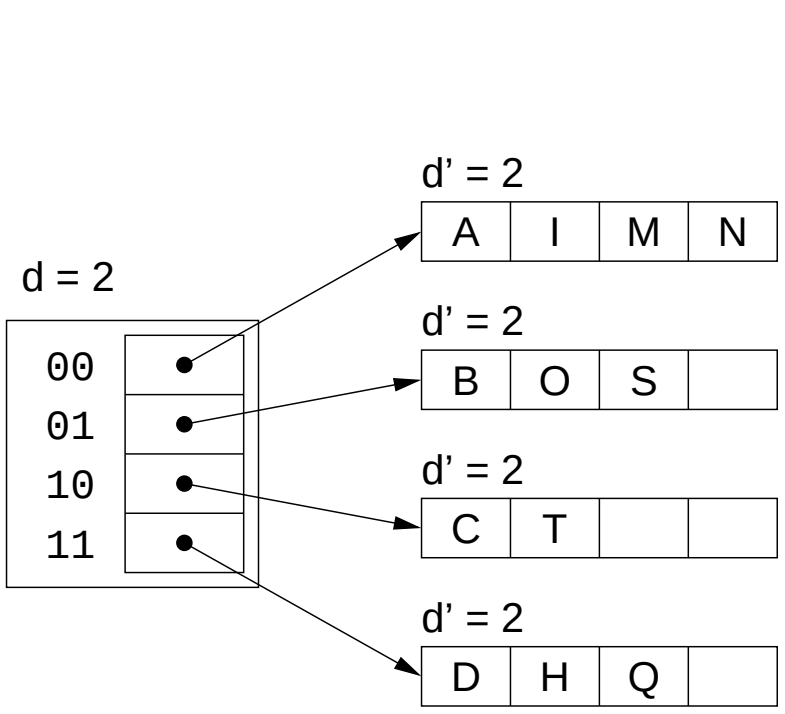
1.1 Kriterien

- Dateiorganisationsform oder Zugriffspfad
- Primärindex oder Sekundärindizes
- Statische oder dynamische Strukturen
- Ein- oder Mehrattributindex, ein- oder mehrdimensionaler Index
- Erhaltung einer Ordnung
- Form der unterstützen Anfragen: Punktanfragen, Bereichsanfragen, partielle Anfragen, vage Anfragen
- Pfadindex: Schemainformation vorhanden oder nicht
- Pfadindex: akkurat, eindeutig, vollständig

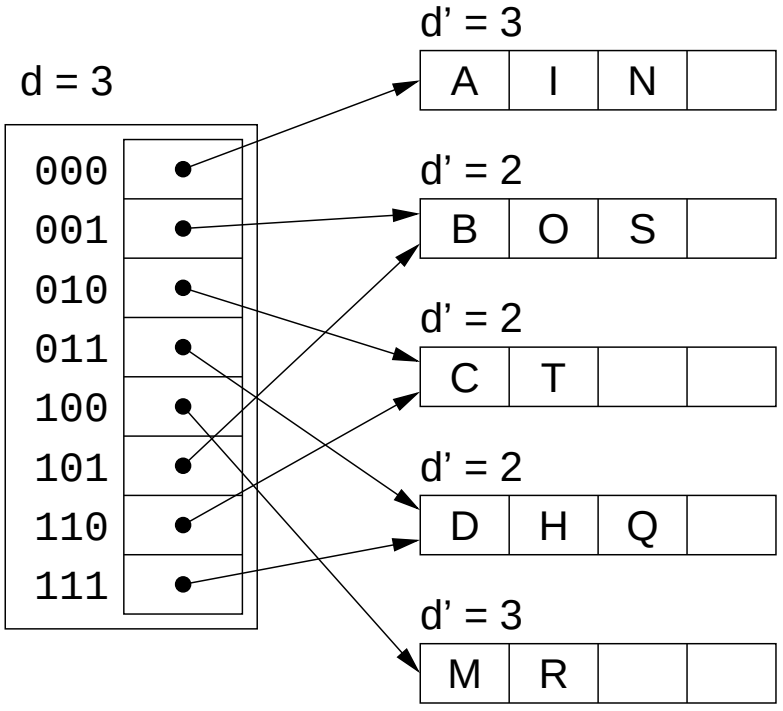
2 Werteindex

- Hash-Strukturen am Beispiel Erweiterbares Hashing
  - Punktanfragen, $O(c)$
- dynamische Index-Bäume am Beispiel B$^+$-Baum
  - Bereichsanfragen, Präfixsuche
- Signatur-Bäume
  - n-dimensionaler Zugriff, Ähnlichkeitssuche

2.1 Erweiterbares Hashing



(a) vor



(b) nach dem Einfügen von „R“

2.2 B$^+$-Baum

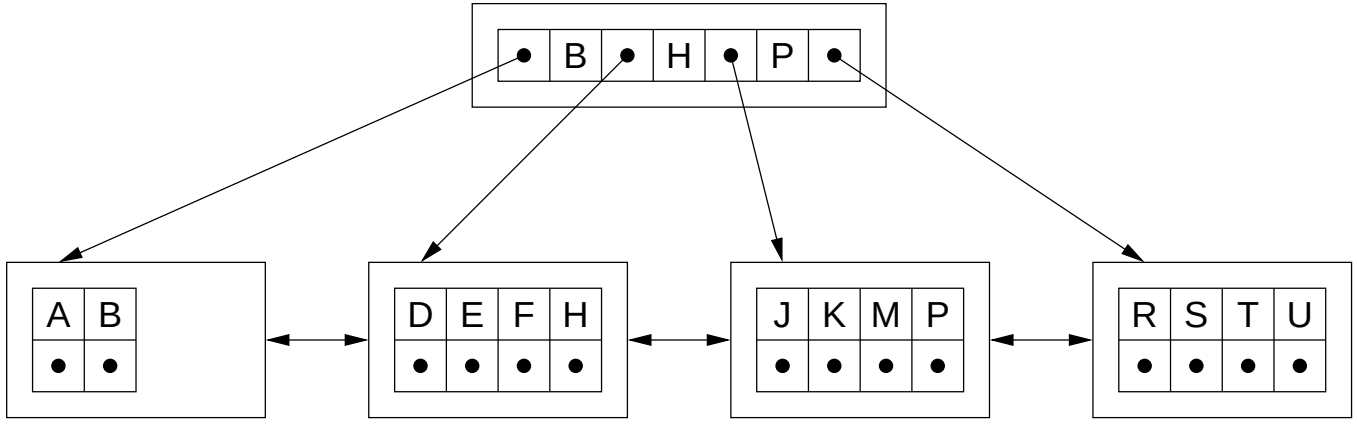


Abbildung 2: B$^+$-Baum, Ordnung $M = 4$

B^+-Baum (cont'd)

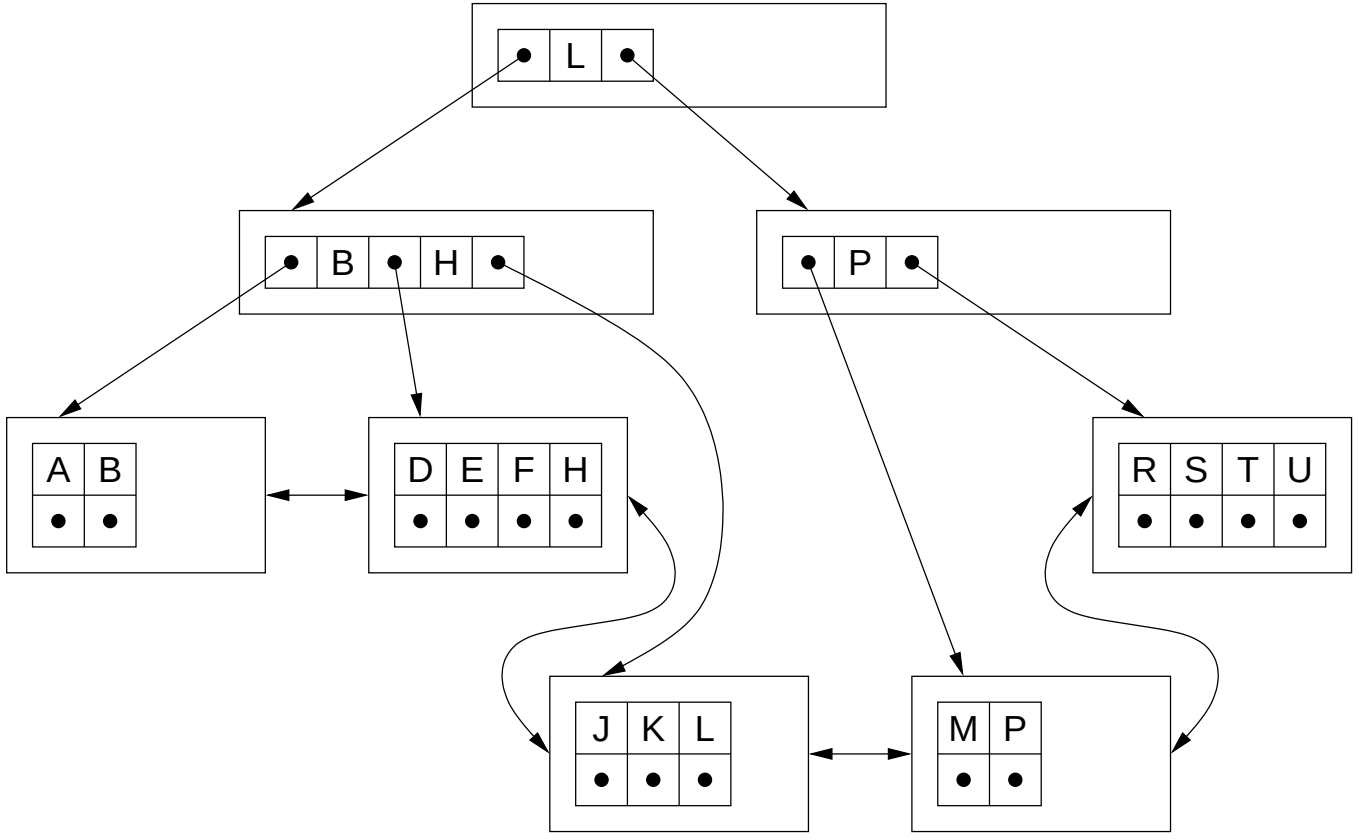


Abbildung 3: nach dem Einfügen von „L“

2.3 Signaturbaum

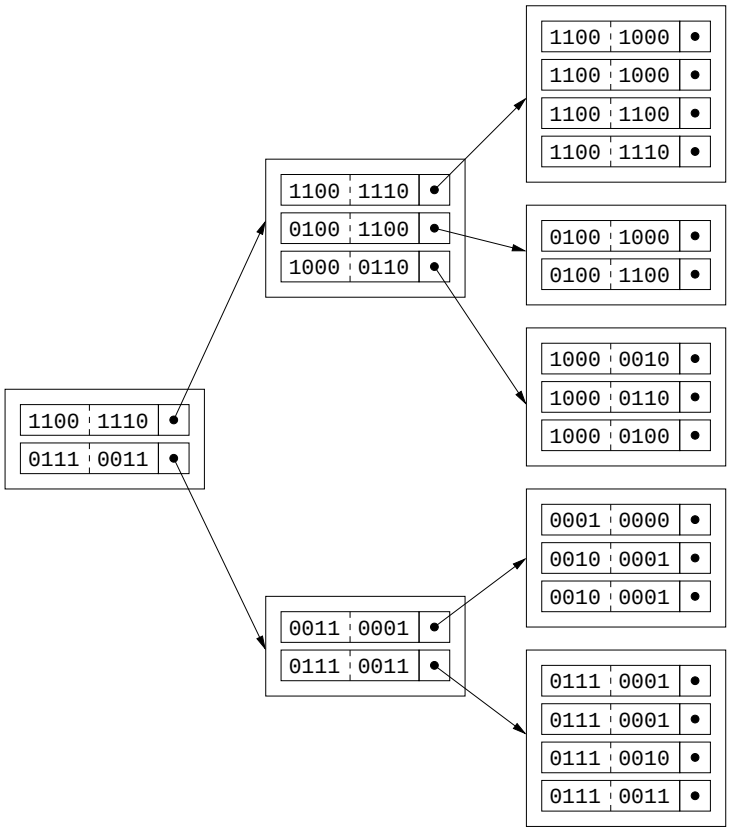


Abbildung 4: Baum, Ordnung $M = 4$ und $m = 2$

Signaturbaum (cont'd)

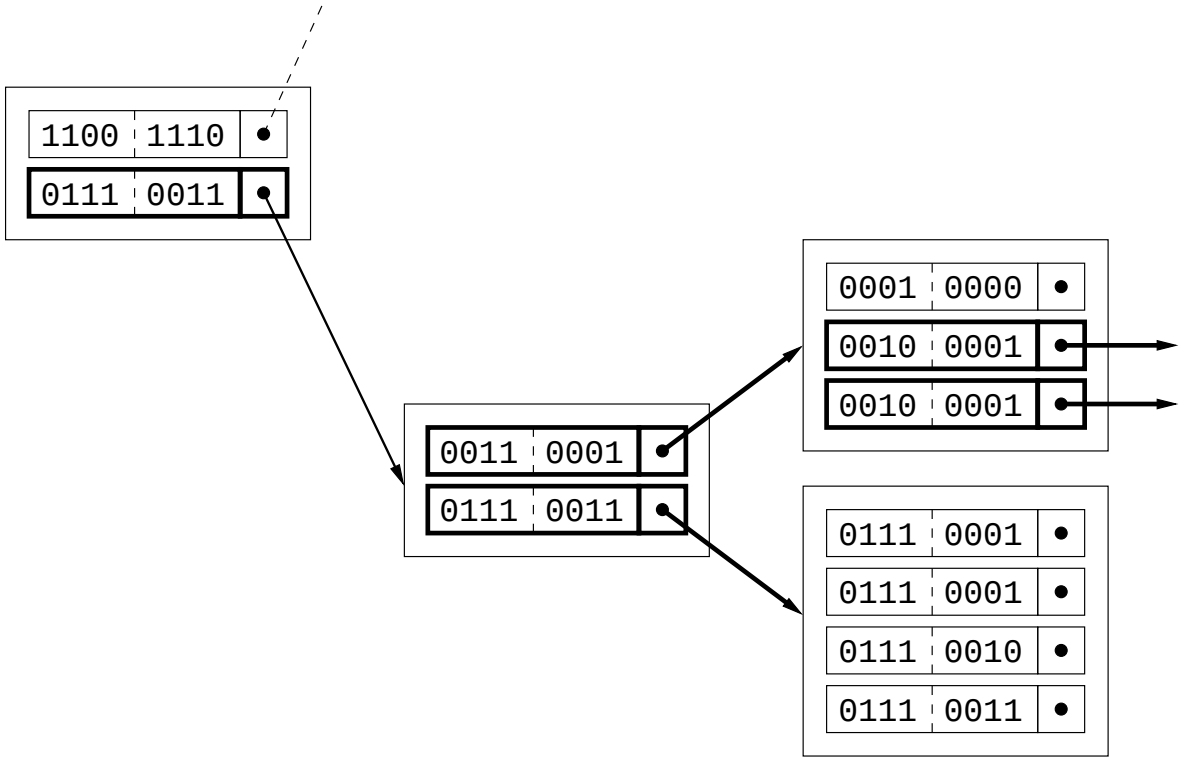


Abbildung 5: Suche nach der Signatur `00100001`

3 Volltextindex

Anfragen an Volltext kann man grob in drei Formen unterteilen:

- Stichwortbasierte Suche,
- Mustersuche und
- Strukturierte Anfragen.

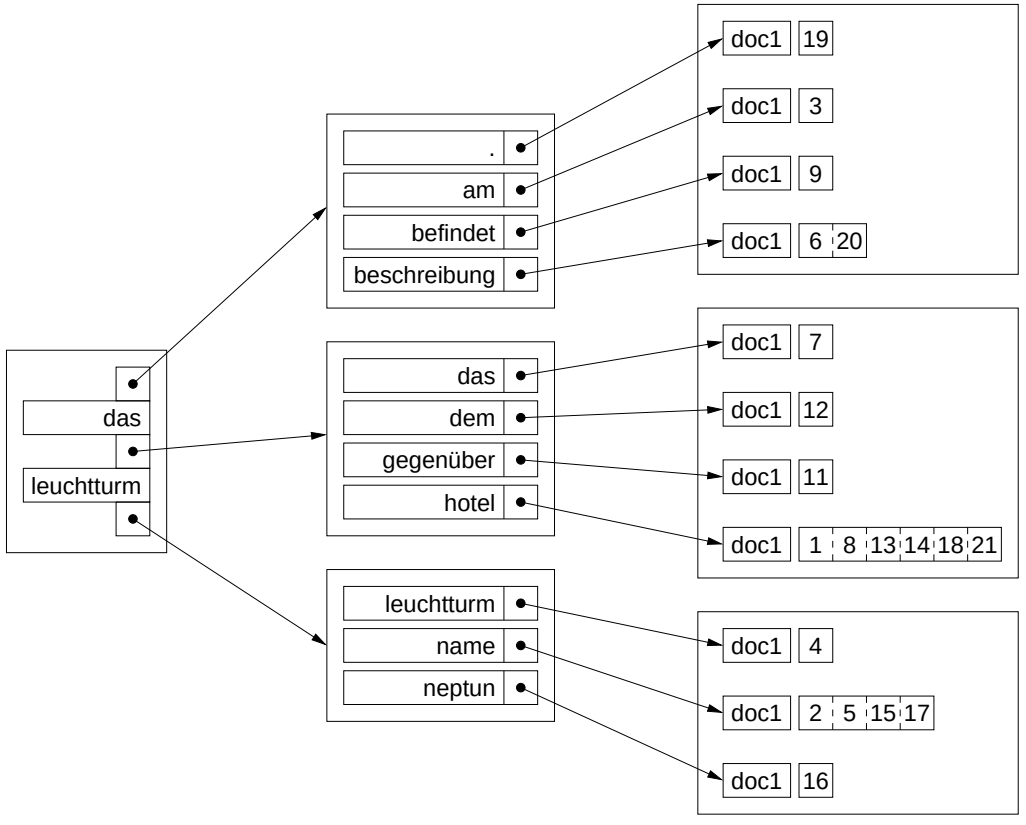
3.1 Invertierte Datei

<hotel[1]>
  <name[2]>Am[3] Leuchtturm[4]</name[5]>
  <beschreibung[6]>
    Das[7] Hotel[8] befindet[9] sich[10] gegenüber[11] dem[12] Hotel[13]
    <hotel[14]><name[15]>Neptun[16]</name[17]></hotel[18]>.[19]
  </beschreibung[20]>
</hotel[21]>

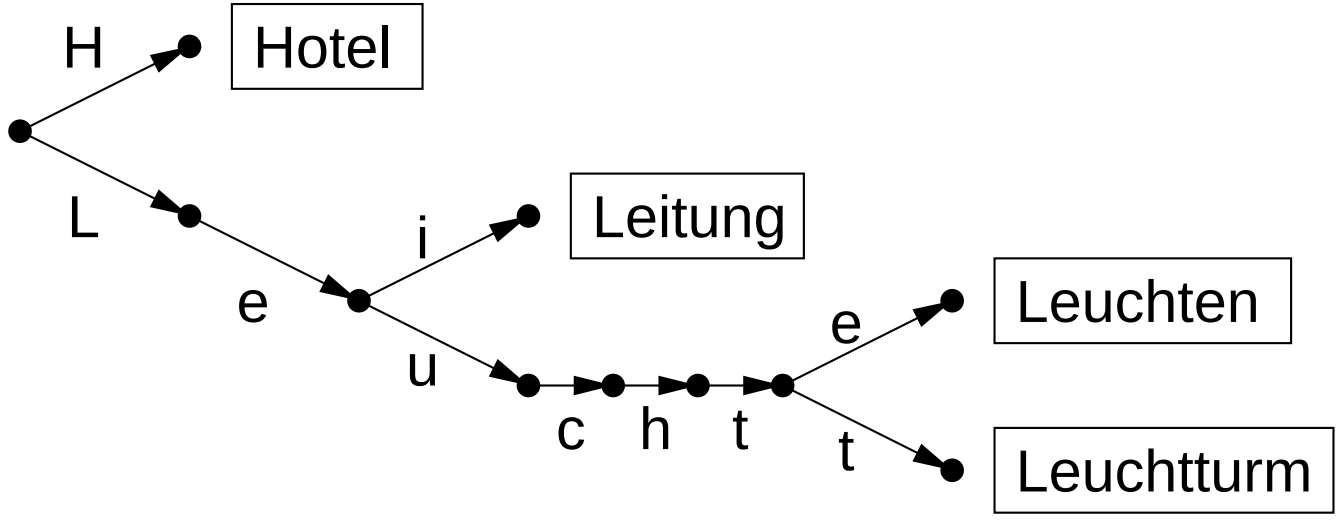
Invertierte Listen — Beispiel

Word	Vorkommen
.	{ 19 }
am	{ 3 }
befindet	{ 9 }
beschreibung	{ 6, 20 }
das	{ 7 }
dem	{ 12 }
gegenber	{ 11 }
hotel	{ 1, 8, 13, 14, 18, 21 }
leuchtturm	{ 4 }
name	{ 2, 5, 15, 17 }
neptun	{ 16 }

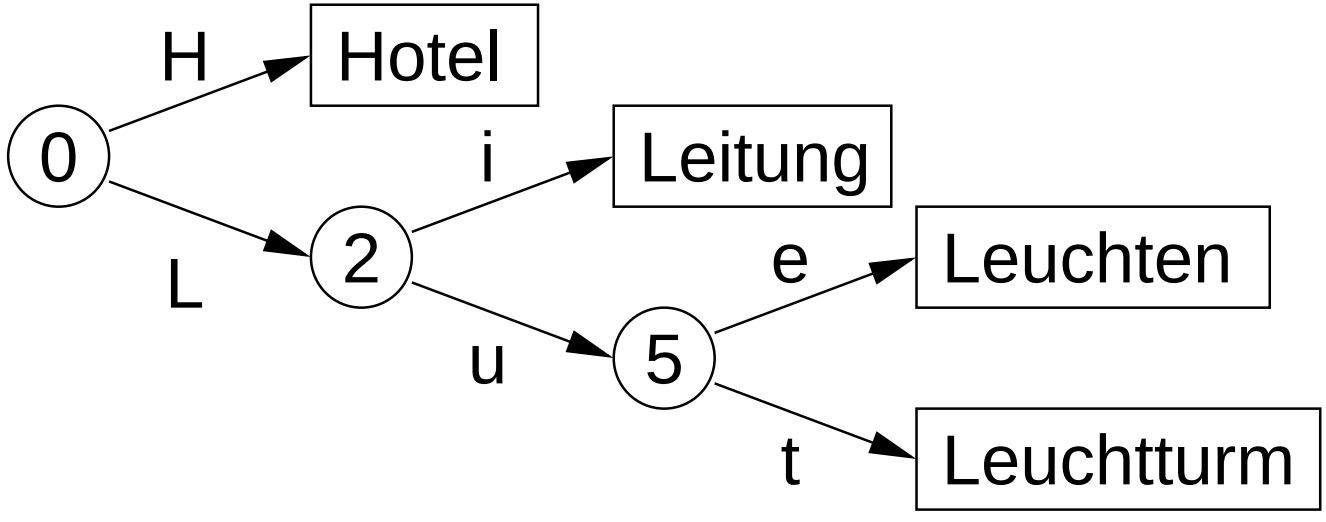
B$^+$-Baum, Ordnung $M = 4$



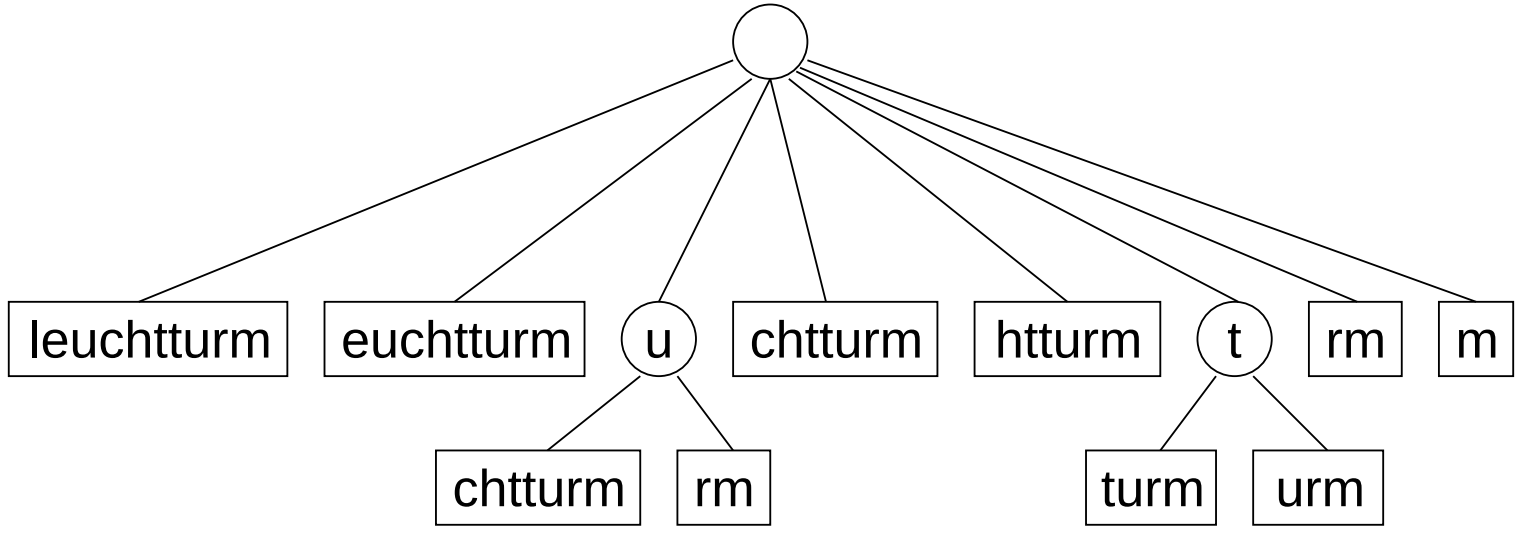
3.2 Tries



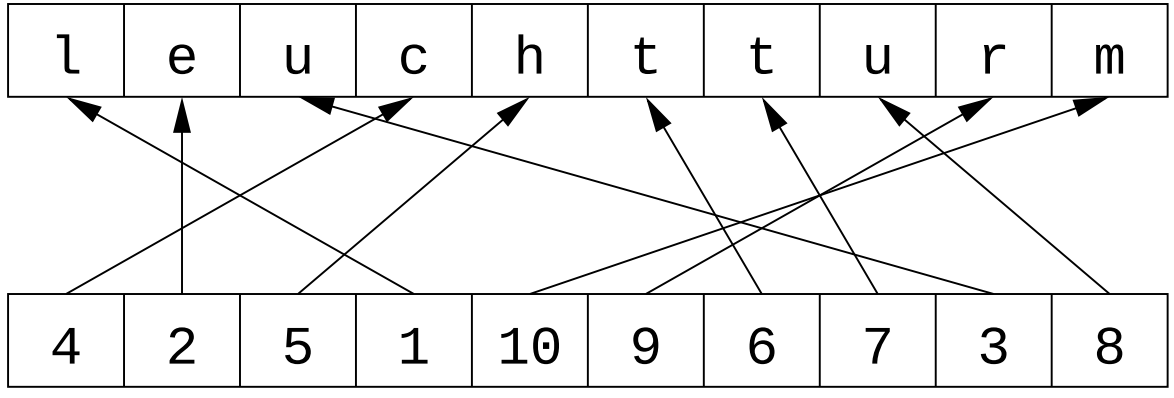
3.3 Patricia-Baum



3.4 Suffixbaum



3.5 Suffix Arrays



Vergleich Volltextindizes

Kriterium	Invert. Datei	Suffix Array	Signatur-baum
exakte Suche	$\surd$	$\surd$	$\surd$
Bereichsanfragen	$\surd$	$\surd$	–
Präfixsuche	$\surd$	$\surd$	–
Teilzeichenketten	–	$\surd$	–
partielle Anfragen	–	$\surd$	$\surd$

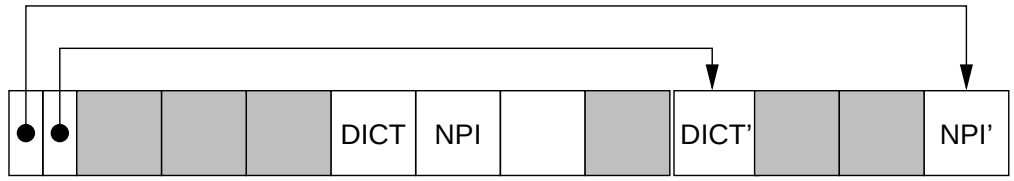
4 Dokumentspeicherung

- komplette physische Speicherung von XML-Dokumenten
- $\longrightarrow$ Speicherung der (DOM-)Knoten, Clustering
- Indizierung der Baum- oder Graphenstruktur eines XML-Dokumentes
- $\longrightarrow$ Pfadindizes, XPath sind nicht General Path Expression (GPE)

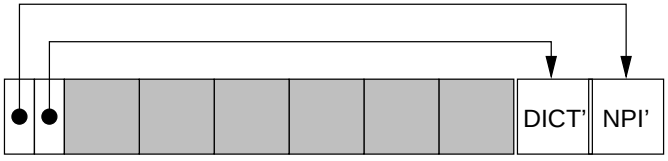
4.1 Persistentes DOM in Infonyte-DB

- persistente Speicherung der durch DOM implizierten Struktur
- pro XML-Dokument eine Datei
- seitenorganisiert: Knotenseite aus 128 DOM-Knoten
- Knotenindex (Node page index, NPI), Elementindex (Dictionary, DICT)
- Strukturindex auf Basis Knotensignatur
- Volltextindex auf Basis invertierter Dateien
- Xpath, XSLT, XQL, XQuery

Aufbau einer *PDOM-Datei*



(a) vor

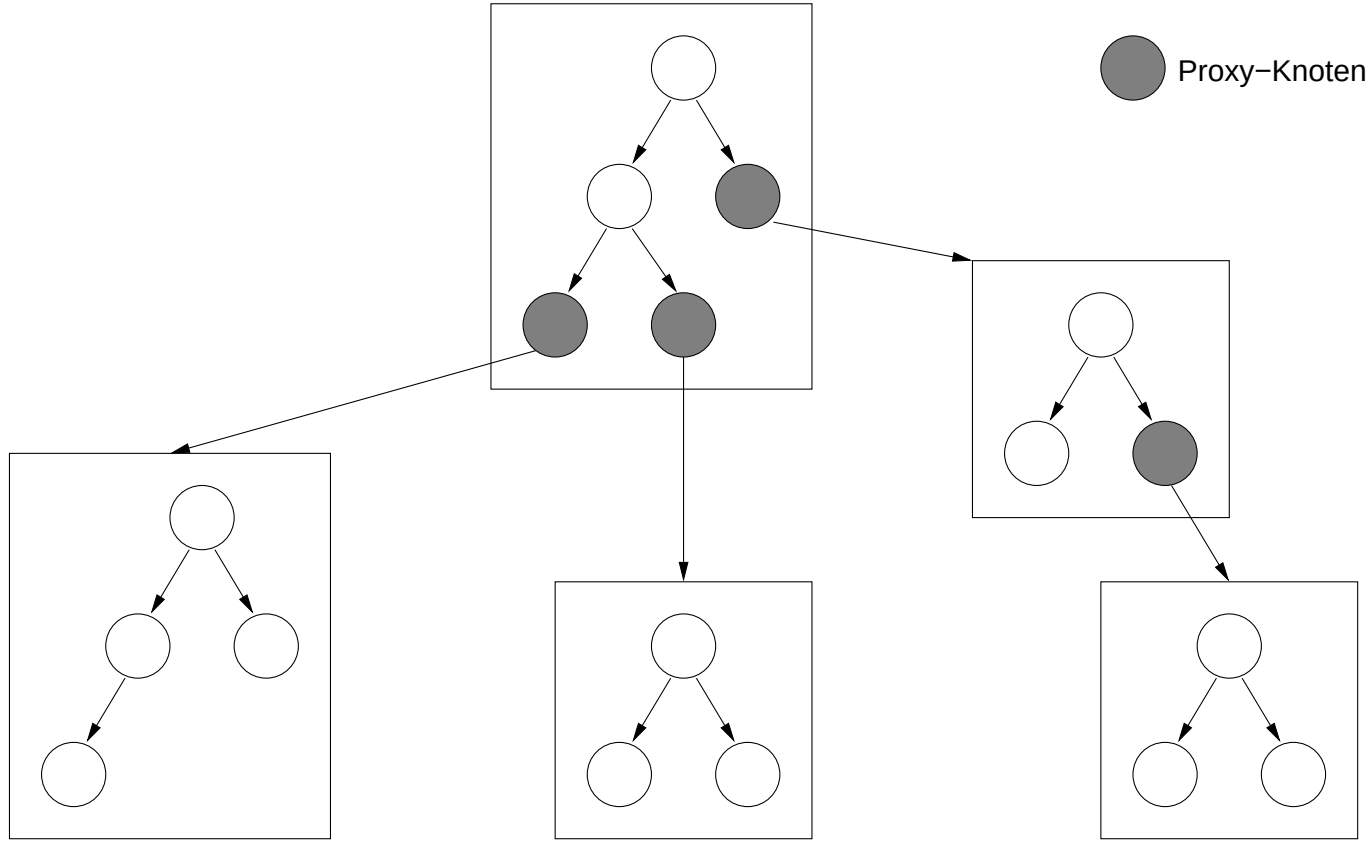


(b) nach Reorganisation

4.2 Hybride Speicherung in Natix

- Knotenstruktur analog DOM,
- seitenbasierte Speicherung,
- Clustering: Teilbäume, Proxy-Knoten,
- B-Baum-ähnliche Struktur, Splitting, Merging

Dokument und Clustering in Natix



5 Struktur-/Pfadindizes

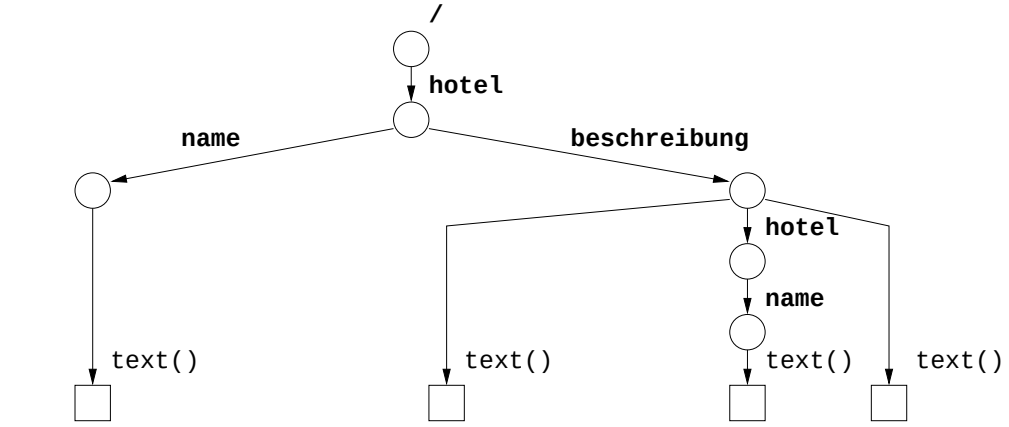
- allgemeine Pfadausdrücke (GPE)
- mit oder ohne Schema (DTD, XML Schema)
- *eindeutig*, jeder Pfad genau einmal
- *akkurat*, alle Pfade exakt abgebildet
- *vollständig*, ganze Pfade, nicht nur Teilpfade

6 Pfadauswertung

- einfache Pfade: `/a/b`, `a/b/c`
- Achsennavigation: Vorgänger, Nachfolger, Geschwister, ...
- einfache Muster: `a//b`, `a/*/b`
- komplizierter: `//` und `*` in Kombination
- allgemein: Pattern matching in Bäumen, reguläre Ausdrücke, NFA ($O(nm)$)

6.1 XML-sensitiver Volltextindex

<hotel[1]>
  <name[2]>Am[3] Leuchtturm[4]</name[5]>
  <beschreibung[6]>
    Das[7] Hotel[8] befindet[9] sich[10] gegenüber[11] dem[12] Hotel[13]
    <hotel[14]><name[15]>Neptun[16]</name[17]></hotel[18]>.[19]
  </beschreibung[20]>
</hotel[21]>

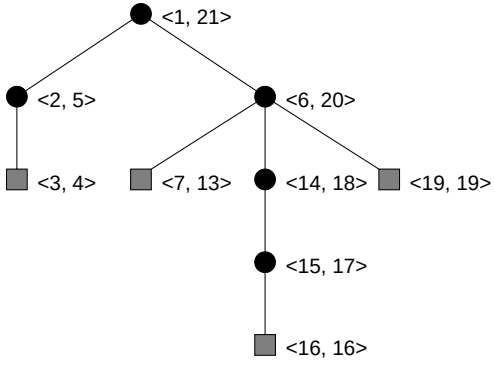


"Das Hotel befindet sich gegenüber dem Hotel"

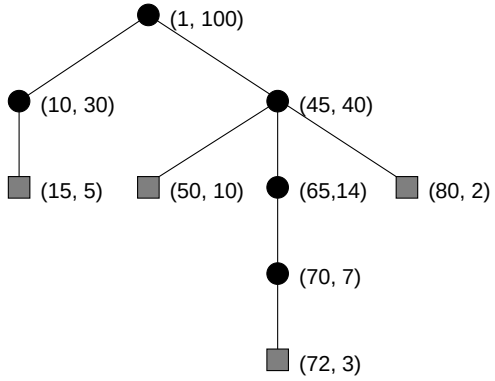
Numerierungsschemata

- Kodierung der Knotenabfolge $\rightarrow$ Achsenavigation, Volltextoperationen
  - Dietz (pre-, post-order), Element-Subelement, Vorgänger/Nachfolger, Phrasen,
  - Position-Size, Updates,
  - Bitvektor, Element-Subelement, Vorgänger/Nachfolger, Subskript, Hierarchie
- Grundlagen: Bereichsalgebren, Suche in metrischen Räumen

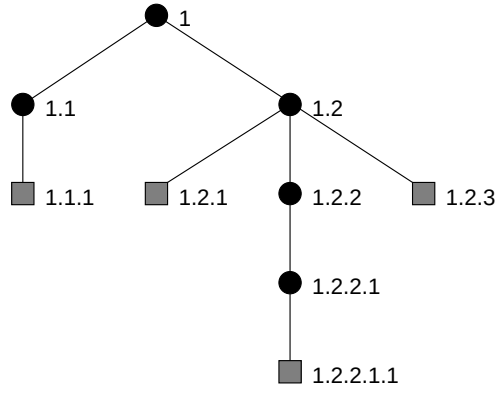
Numerierungsschemata (cont'd)



(c) Dietz

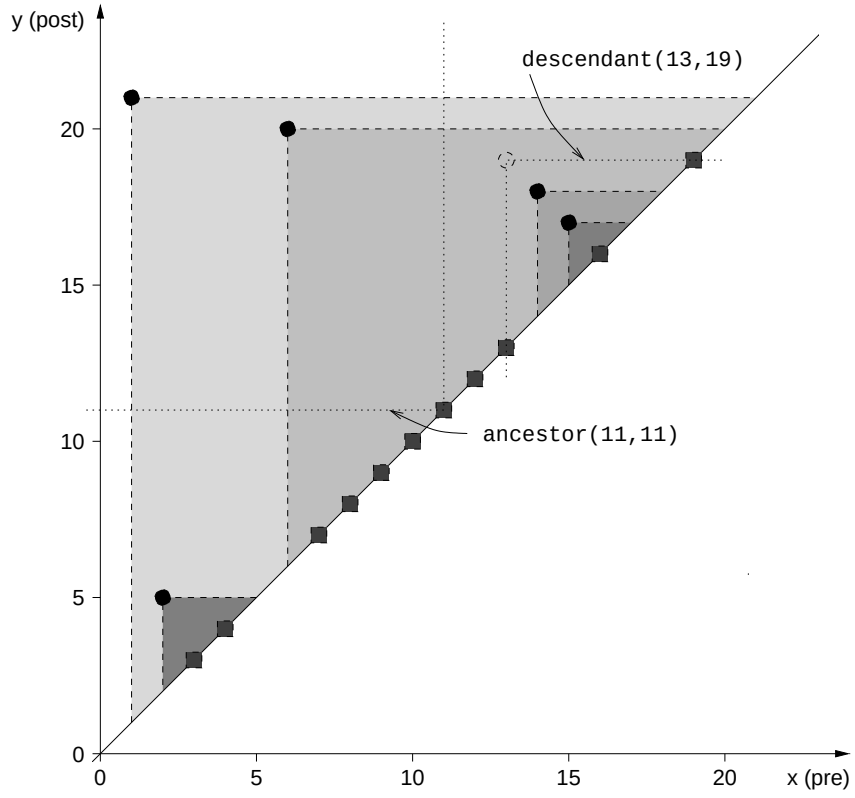


(d) Size



(e) Bitvektor

Numerierungsschemata (cont'd)



Elementindizierung

- Wortposition (*Start-Tag, Ende-Tag*)
- **CONTAINS**-Anfragen, SQL/MM Fulltext
- einfache Pfadanfragen, Element-Subelement
- Aufbau analog Volltextindex mit Wortpositionen

Element	Vorkommen
beschreibung	$\{(6, 20)\}$
hotel	$\{(1, 21), (14, 18)\}$
name	$\{(2, 5), (15, 17)\}$

Anfragen

Anfragetyp: *elem//subelem*

```
select sub.*
from element e, element sub
where e.begin < sub.begin
  and sub.end < e.end
  and e.name = elem
  and sub.name = subelem
```

6.2 XASR

- Pfadindizierung, Pre-Post-Order, Vaterknoten,
- Element-Subelement-Anfragen $\rightarrow$ Verbundfolgen

docid	position		dparent	element
	dmin	dmax		
4711	1	21	0	hotel
4711	2	5	1	name
4711	6	20	1	beschreibung
4711	14	18	6	hotel
4711	17	17	14	name

Anfragen

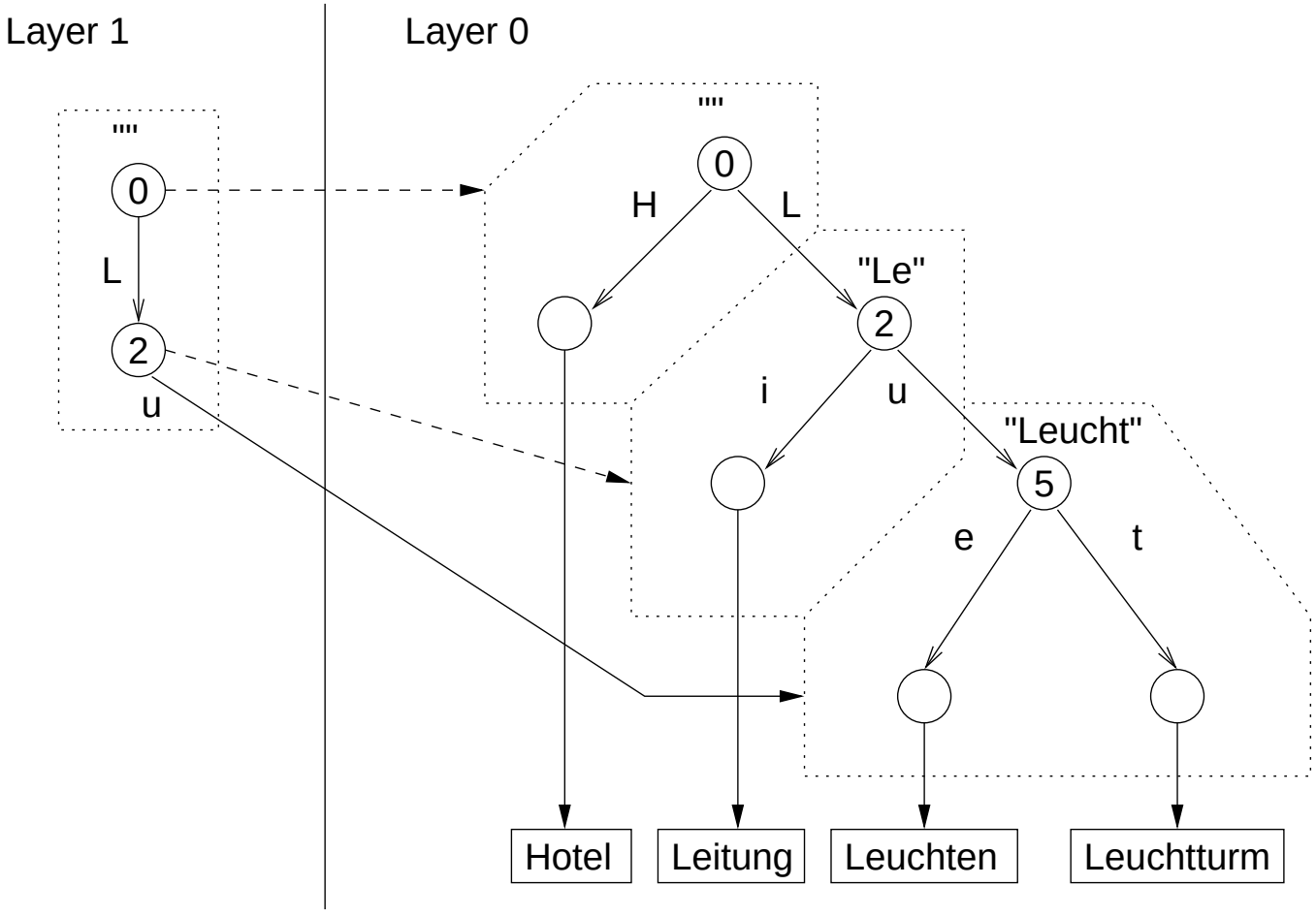
Anfragetyp: *elem/subelem*

```
select sub.*
from element e, element sub
where e.dmin = sub.dparent
  and e.name = elem
  and sub.name = subelem
```

6.3 Index Fabric

- Vollständige Pfade, ausgewählte Teilpfade für häufige Anfragen
- akkurat, eindeutig
- kein Schema
- Mehr-Ebenen-Patricia-Baum
- spezielle Elementkodierung
  - Elementnamen als Zeichen eines erweiterten Alphabets
  - Präfixnotation
  - Beispiel Hotelnamen: `hotel` $\rightarrow$ **`H`**, `name` $\rightarrow$ **`N`**
    - „**`H N`** `Am Leuchtturm`“
    - „**`H N`** `Neptun`“

Mehr-Ebenen-Patricia-Baum



6.4 Pfad-Indizierung in SphinX

- Schema notwendig
- Schemagraph und B^+-Baum
- akkurat, eindeutige Pfadindizierung
- GPE

SphinX Index-Struktur

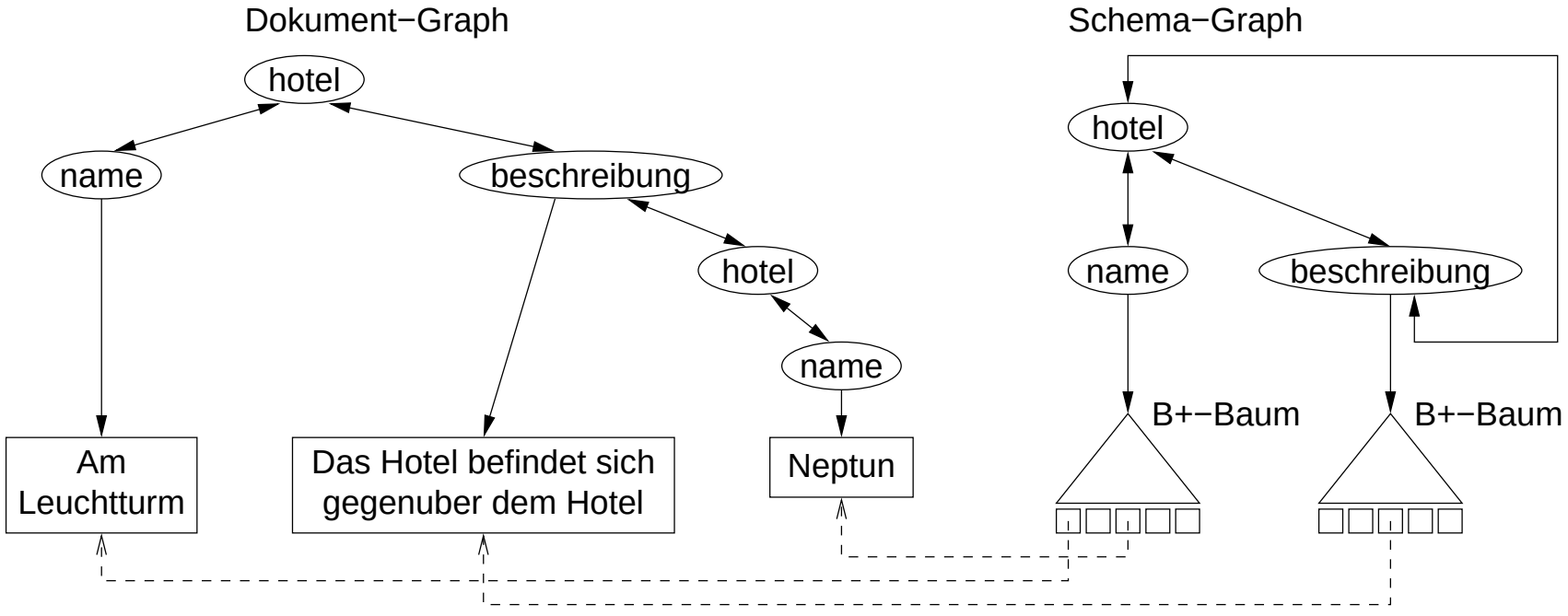


Abbildung 6: Dokument- und Schemagraph

7 Zusammenfassung

- Vielzahl weiterer Techniken: Dataguides, 1-Index, 2-Index, T-Index, ToXin, . . .
- Unterstützung nicht nur für einfache Pfadausdrücke, sondern GPE
- Aufwand für GPE: analog zu regulären Ausdrücken, NFA, DFA
- Verbesserung wenn Schema vorhanden (Dokument- vs. Schemagraph)