

**Übung zur Vorlesung
„Einsatz und Realisierung von Datenbanksystemen“
im Sommersemester 2007**

Richard Kuntschke (richard.kuntschke@in.tum.de)

Lösungen zu Blatt 12

Aufgabe 1

Abbildung 1 zeigt das Beispiel-Dokument. Die Knoten wurden hierarchisch nummeriert. Jede Knotenmarkierung (auch *ORDpfad*-Kennung genannt) gibt die Position des Knotens eindeutig wieder, so dass das ursprüngliche XML-Dokument wieder rekonstruiert werden kann. So lässt sich z.B. aus der Nummer 1.3.2 sofort ableiten, dass der zugehörige Vaterknoten die *ORDpfad*-Kennung 1.3 hat. Die Nummerierung gibt sehr gut den Aufbau des XML-Dokuments wieder, weist jedoch Nachteile bei Änderungen auf. Möchte man etwa nachträglich Thomas Dürrenmatt und Friedrich Mann als zwei weitere Autoren des Buchs Datenbanksysteme zwischen Alfons Kemper und André Eickler einfügen, so müssen der Knoten mit der Kennung 1.3.2 und alle seine Kindknoten neu nummeriert werden. Er erhält dann die *ORDpfad*-Kennung 1.3.4. Die Kinder werden entsprechend mit den Kennungen 1.3.4.1 und 1.3.4.2 versehen. Abhängig von der Größe des Teilbaums, den der Knoten mit der Marke 1.3.2 hat, kann eine einfache Einfügeoperation also schnell aufwändig werden. Gewünscht sind folglich Nummerierungsverfahren, die Updates gut unterstützen, also keine oder wenige (bzw. seltene) Umnummerierungen erfordern.

Reservieren von Einfügestellen Eine einfache Möglichkeit, Erweiterungen eines gegebenen XML-Dokuments besser zu unterstützen, ist es, Platz für weitere potentielle Einfügeoperationen zu reservieren. Wie in der Aufgabenstellung angemerkt, kann man zwischen Geschwisterknoten

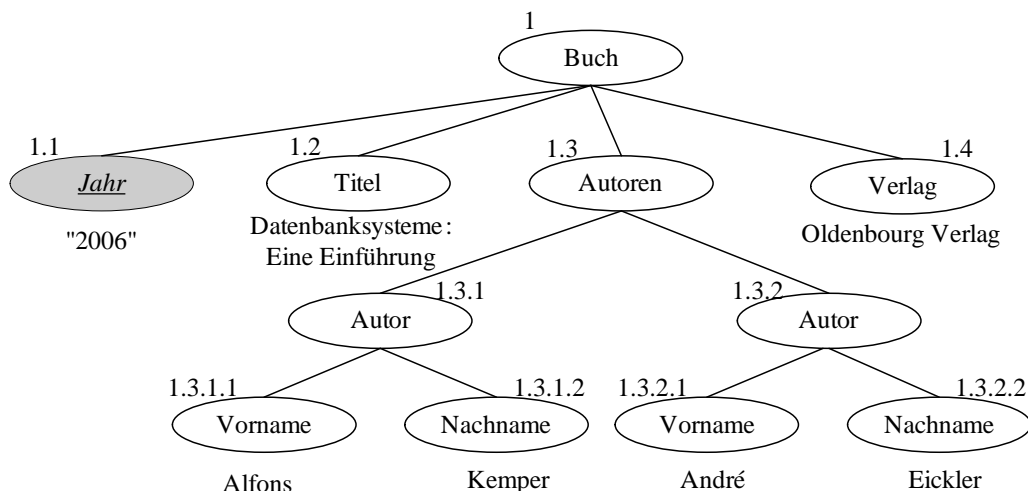
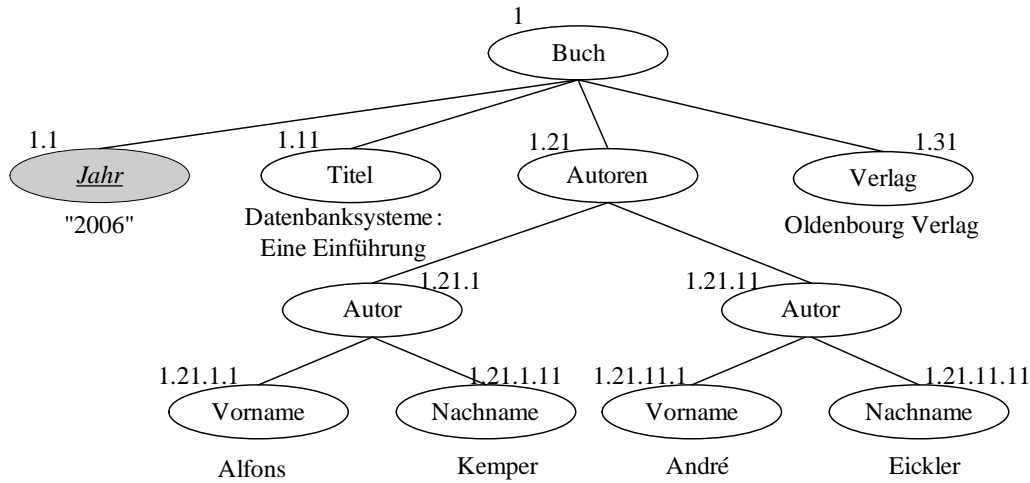
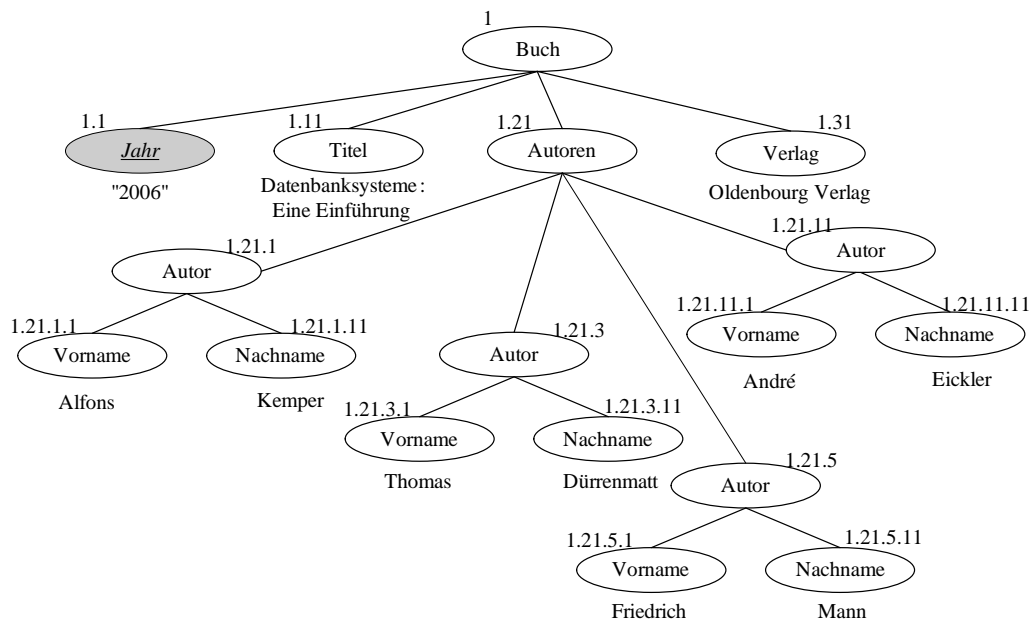


Abbildung 1: Hierarchische Nummerierung von Knoten in einem XML-Dokument



(a) Nummerierung, die einen Freiraum zwischen Geschwisterknoten vorsieht



(b) Baum nach dem Einfügen zweier zusätzlicher Autoren

Abbildung 2: Hierarchische Nummerierung von Knoten mit Puffer für Einfügestellen

beispielsweise Platz für weitere Knoten vorsehen, indem man die Geschwister nicht zusammenhängend durchnummeriert, sondern etwa in Abständen von jeweils 10. Dies ist in Abbildung 2(a) für die Baumdarstellung des bekannten XML-Dokuments gezeigt.

Die Autoren Thomas Dürrenmatt und Friedrich Mann können nun problemlos zwischen Alfons Kemper und André Eickler in das XML-Dokument eingefügt werden, ohne dass eine Umnummerierung bestehender Knoten erforderlich ist. Fügt man die *Autor*-Elementknoten zwischen die Knoten mit den Markierungen 1.21.1 und 1.21.11 ein, so kann man ihnen beispielsweise die Label 1.21.2 (für Thomas Dürrenmatt) und 1.21.3 (für Friedrich Mann) zuweisen. Alternativ kann man auch wieder Platz für weitere Einfügestellen reservieren, wie dies in Abbildung 2(b) angedeutet ist.

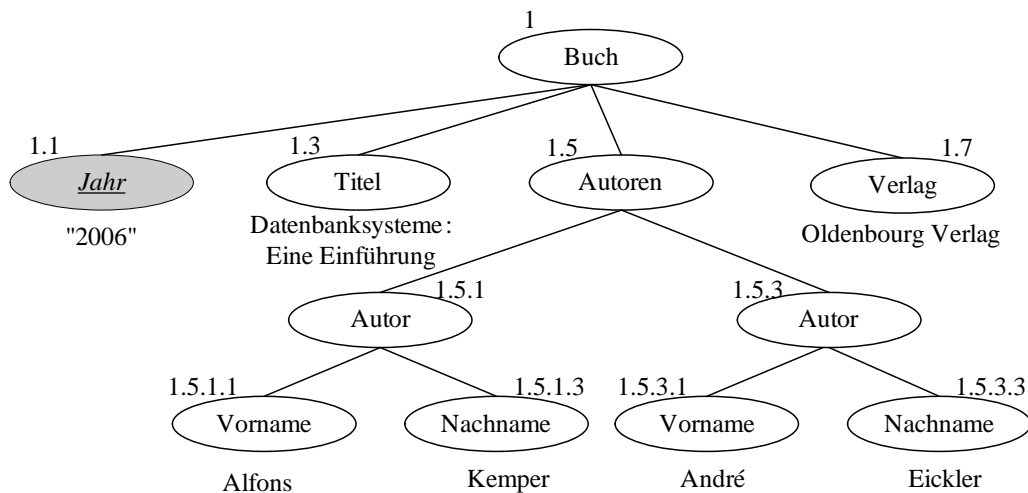
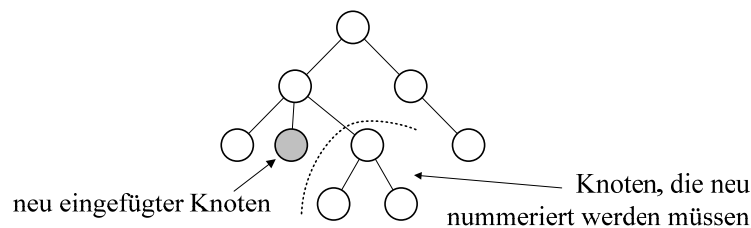


Abbildung 3: „Einfüge-freundliche“ hierarchische Nummerierung von Knoten

Diese Vorgehensweise ist prinzipiell flexibler als die ursprüngliche Nummerierung, bei der keine Zwischenräume vorgesehen waren. Allerdings ist diese Flexibilität von der Größe der Freiräume abhängig. Treten relativ viele Updates auf, bzw. ist die Größe des vorgesehenen Puffers zu klein, so sind auch in diesem Fall Umnummerierungen notwendig. Nachfolgende Darstellung zeigt allgemein, dass beim Einfügen eines neuen Knotens in einen Baum die rechten Geschwisterknoten neu nummeriert werden müssen, falls der Zwischenraum für weitere Knotennummierungen aufgebraucht ist:

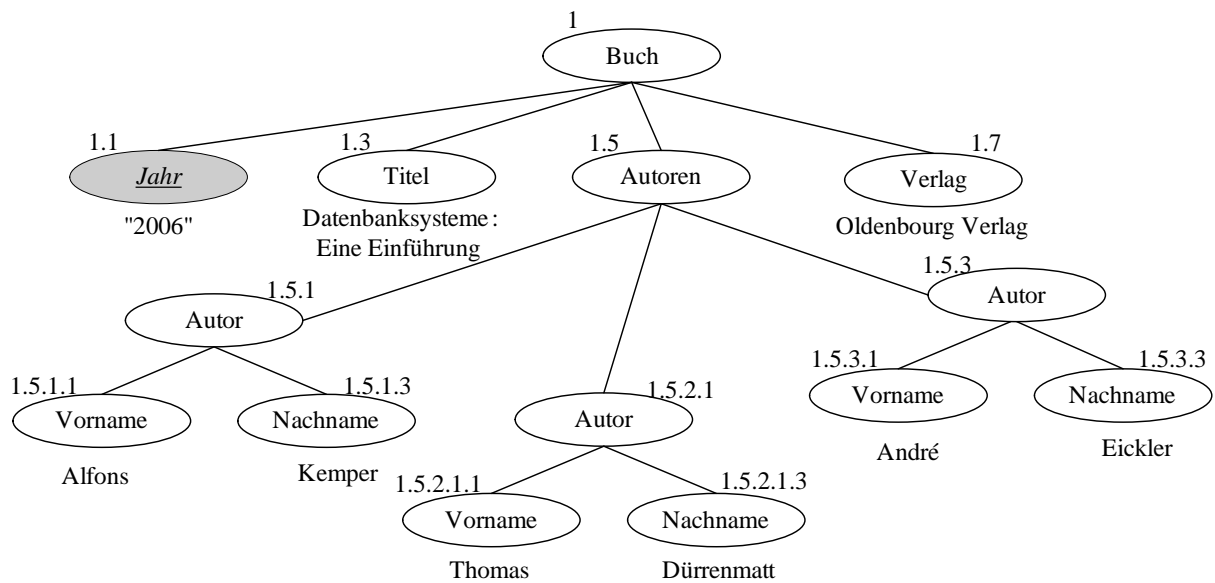


Das Verfahren schließt Umnummerierungen somit nicht generell aus, sondern verringert nur die Wahrscheinlichkeit dafür. Diese Wahrscheinlichkeit verringert sich, je größer die Zwischenräume gewählt werden. Der Nachteil großer Zwischenräume ist jedoch, dass die Knotenmarkierungen entsprechend länger werden und somit mehr Speicherplatz für die Kodierung benötigt wird.

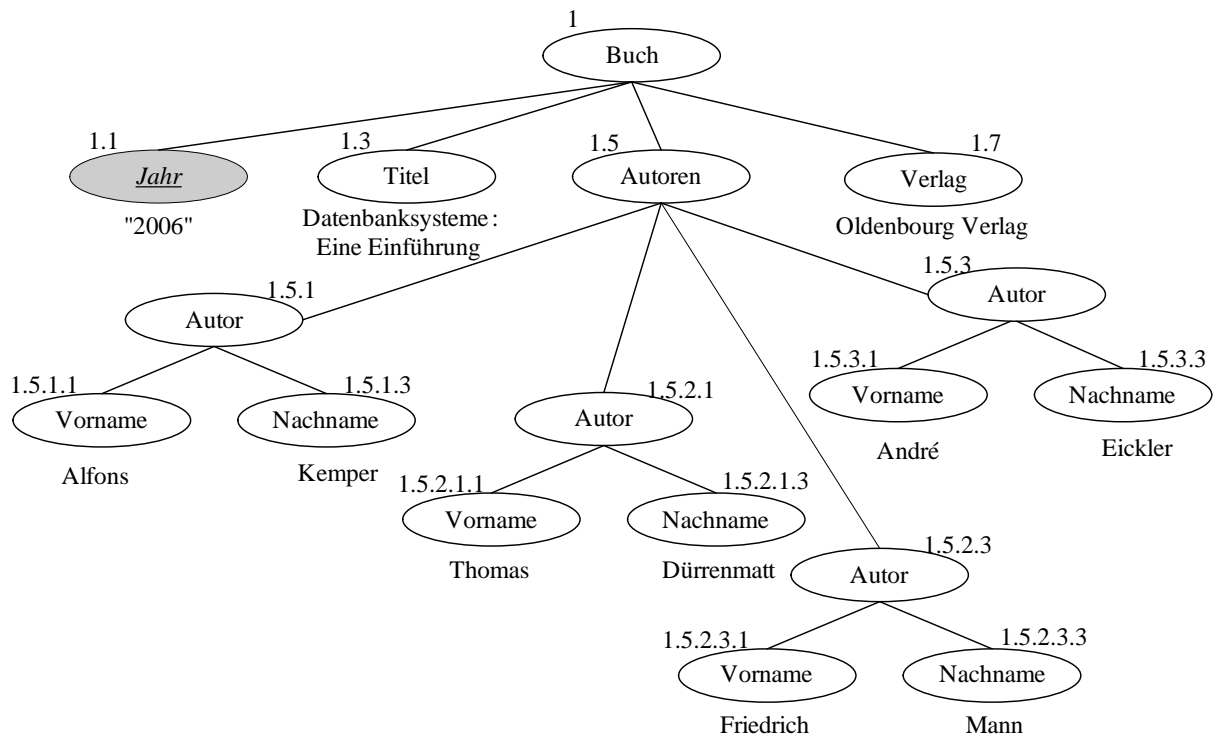
Nach wie vor unterstützt diese Nummerierung die bekannte Methode zum Bestimmen des Vaterknotens. Ist $X.y$ die *ORDpfad*-Kennung eines Knotens – wobei y die letzte Komponente der Nummerierung ist –, so ist X die *ORDpfad*-Kennung des Vaterknotens.

„Einfüge-freundliche“ Knotenmarkierungen Wir wollen nun eine Knotennummerierung vorstellen, die Änderungsoperationen sehr effizient unterstützt. Im Gegensatz zur zuvor betrachteten Methode ist es bei diesem Verfahren nicht erforderlich, vorhandene Knoten nach Einfügevorgängen neu zu nummerieren.

Wie in der Aufgabenstellung angegeben, fordern wir, dass die letzte Komponente einer *ORDpfad*-Kennung stets ungerade ist. Ganzzahlige Nummern verwenden wir, um die Einfügeposition nachträglich eingefügter Knoten zu markieren. Abbildung 3 zeigt die Nummerierung der Knoten



(a) Baum nach dem Einfügen von Thomas Dürrenmatt



(b) Baum nach dem Einfügen von Friedrich Mann

Abbildung 4: Hierarchische Nummerierung von Knoten mit Puffer für Einfügestellen

für das ursprüngliche XML-Dokument.

Wollen wir nun einen weiteren *Autor*-Elementknoten für Thomas Dürrenmatt einfügen, so gehen wir wie folgt vor: Der zusätzliche Knoten soll zwischen die Knoten mit den *ORDpfad*-Kennungen 1.5.1 und 1.5.3 eingefügt werden. Deshalb weisen wir dem neuen Knoten die *ORD*-

pfad-Kennung 1.5.2.1 zu. Durch den Präfix 1.5.2 wird ausgedrückt, dass der neue Knoten zwischen den beiden vorhandenen *Autor*-Knoten eingefügt ist. Die Kindknoten (*Vorname* und *Nachname*) werden wie gehabt durchnummeriert. Abbildung 4(a) zeigt den Dokumentbaum nach dem beschriebenen Einfügevorgang.

In analoger Weise fügen wir die Daten des zusätzlichen Autors Friedrich Mann in das Dokument ein. Der entsprechende *Autor*-Knoten bekommt die *ORDpfad*-Kennung 1.5.2.3. Abbildung 4(b) zeigt den Baum nach dem Einfügevorgang. Die angepasste *InfoTab*-Ausprägung ist in Abbildung 5 gezeigt, wobei die zusätzlich eingefügten Einträge grau hinterlegt dargestellt sind.

Der Vaterknoten eines Knotens mit bekannter *ORDpfad*-Kennung x wird wie folgt berechnet:

1. entferne die letzte Komponente von x
2. entferne alle geradzahigen Komponenten am Ende der Kennung.

Beispiele: Für $x = 1.5.3$ ist die Kennung des Vaterknotens 1.5. Für $x = 1.5.2.1$ bestimmt sich die Nummer des Vaterknotens ebenfalls zu 1.5.

[Tatarinov et al., 2004] stellen einen Überblick über Techniken zur Nummerierung von XML-Knoten zur Speicherung von XML-Dokumenten in relationalen Datenbanksystemen bereit.

Das zuletzt beschriebene Nummerierungsverfahren wurde von [O’Neil et al., 2004] vorgestellt und ist in SQL Server 2005 implementiert. Die Autoren gehen in dem Beitrag auch auf Kodierungsverfahren zur effizienten Speicherung der *ORDpfad*-Kennungen ein.

Aufgabe 2

Dieses Verfahren basiert auf einer geschickten und effektiven Nummerierung der Knoten eines XML-Dokuments. Jedem Knoten werden zweidimensionale Koordinaten bestehend aus Preorder- und Postorder-Nummerierung zugeordnet.

Die Preorder eines Knotens wird aufsteigend in der Reihenfolge der öffnenden Elementtags vergeben, die Postorder aufsteigend in Reihenfolge der schließenden Elementtags. Sei also *pre* der Zähler zur Bestimmung der Preorder und *post* der Zähler für die Bestimmung der Postorder. Liest man den öffnenden Elementtag des Knotens a , d.h. $\langle a \rangle$, so bekommt a die Preorder $++pre$ (d.h. *pre* wird inkrementiert). Liest man den schließenden Tag $\langle /a \rangle$, erhält der Knoten a die Postorder $++post$.

Wir wollen diese Nummerierung an einem einfachen abstrakten Beispiel – das sich noch nicht auf den in Abbildung 6 dargestellten Baum bezieht – zeigen. Folgende Tabelle skizziert ein relativ simples XML-Dokument. Um die Nummerierung zu erzeugen, muss man das XML-Dokument nur einmal sequentiell durchlaufen¹.

Dokument	Operation	<i>pre</i>	<i>post</i>
$\langle a \rangle$	Preorder von a ist 1	1	0
$\langle b \rangle$	Preorder von b ist 2	2	0
$\langle /b \rangle$	Postorder von b ist 1	2	1
$\langle c \rangle$	Preorder von c ist 3	3	1
$\langle d \rangle$	Preorder von d ist 4	4	1
$\langle /d \rangle$	Postorder von d ist 2	4	2
$\langle /c \rangle$	Postorder von c ist 3	4	3
$\langle /a \rangle$	Postorder von a ist 4	4	4

¹Das Ermitteln der Nummerierung entspricht einem Durchlaufen des Graphen gemäß dem Eulerpfad. Dabei muss man sich nicht alle zuvor gelesenen Elemente merken. Sobald ein Element geschlossen wird, kann man es aus dem lokalen Speicher entfernen. Wenn h die Höhe des XML-Dokuments in Baumdarstellung ist, so wird Speicherplatz für maximal h Elemente benötigt. Damit kann diese Nummerierung auch für große (nicht entartete) Dokumente effizient berechnet werden.

InfoTab						
DocID	ORDpfad	Tag	KnotenTyp	Wert	Pfad	invPfad
4711	1	Buch	Element	-	#Buch	#Buch
4711	1.1	Jahr	Attribut	2006	#Buch#@Jahr	#@Jahr#Buch
4711	1.3	Titel	Element	Datenbank...	#Buch#Titel	#Titel#Buch
4711	1.5	Autoren	Element	-	#Buch#Autoren	#Autoren#Buch
4711	1.5.1	Autor	Element	-	#Buch#Autoren#Autor	#Autor#Autoren#Buch
4711	1.5.1.1	Vorname	Element	Alfons	#Buch#Autoren#Autor#Vorname	#Vorname#Autor#Autoren#Buch
4711	1.5.1.3	Nachname	Element	Kemper	#Buch#Autoren#Autor#Nachname	#Nachname#Autor#Autoren#Buch
4711	1.5.2.1	Autor	Element	-	#Buch#Autoren#Autor	#Autor#Autoren#Buch
4711	1.5.2.1.1	Vorname	Element	Thomas	#Buch#Autoren#Autor#Vorname	#Vorname#Autor#Autoren#Buch
4711	1.5.2.1.3	Nachname	Element	Dürrenmatt	#Buch#Autoren#Autor#Nachname	#Nachname#Autor#Autoren#Buch
4711	1.5.2.3	Autor	Element	-	#Buch#Autoren#Autor	#Autor#Autoren#Buch
4711	1.5.2.3.1	Vorname	Element	Friedrich	#Buch#Autoren#Autor#Vorname	#Vorname#Autor#Autoren#Buch
4711	1.5.2.3.3	Nachname	Element	Mann	#Buch#Autoren#Autor#Nachname	#Nachname#Autor#Autoren#Buch
4711	1.5.3	Autor	Element	-	#Buch#Autoren#Autor	#Autor#Autoren#Buch
4711	1.5.3.1	Vorname	Element	André	#Buch#Autoren#Autor#Vorname	#Vorname#Autor#Autoren#Buch
4711	1.5.3.3	Nachname	Element	Eickler	#Buch#Autoren#Autor#Nachname	#Nachname#Autor#Autoren#Buch
4711	1.7	Verlag	Element	Oldenbourg V...	#Buch#Verlag	#Verlag#Buch

Abbildung 5: InfoTab-Ausprägung nach dem Einfügen von Thomas Dürrenmatt und Friedrich Mann

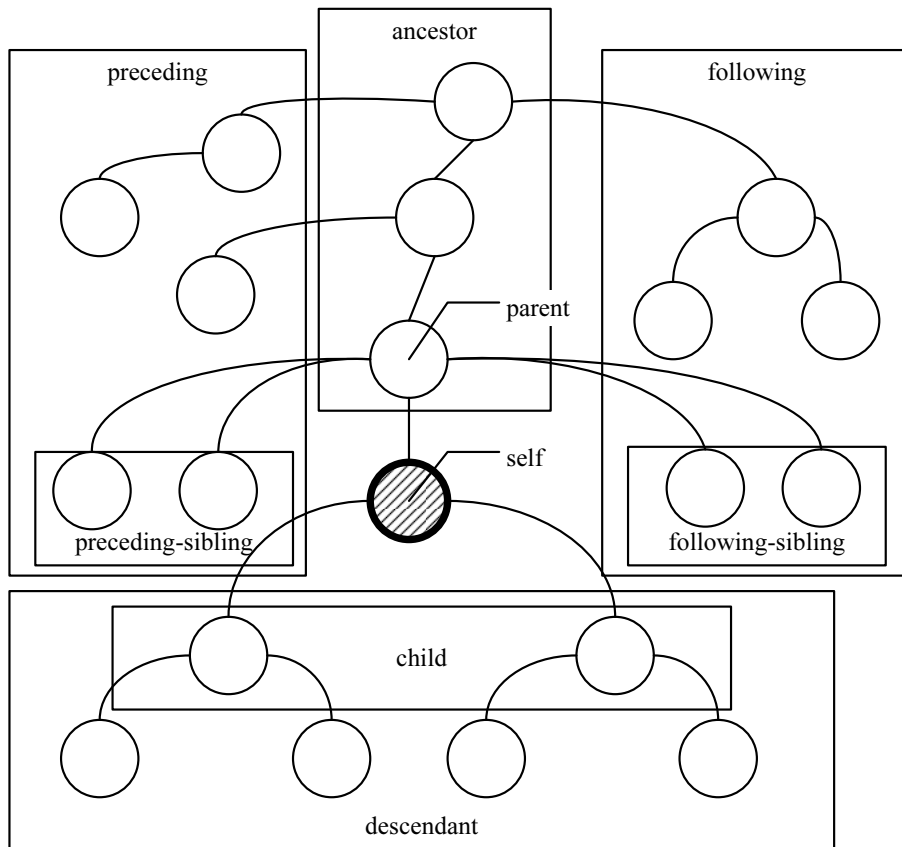


Abbildung 6: Visualisierung der XPath-Pfadausdrücke

Die folgende Abbildung zeigt die graphische Repräsentation des Baums: Links ist der Elementbaum dargestellt, rechts der Baum mit (*Preorder*, *Postorder*)-Nummerierung, wobei links oberhalb eines jeden Knotens die Preorder-Nummerierung und rechts unterhalb die Postorder-Nummerierung angegeben ist.

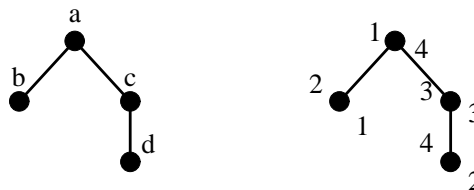


Abbildung 7 zeigt den in der Aufgabenstellung angegebenen Graphen. In der Abbildung wurden die Knoten von *a* bis *t* benannt. Die Preorder ist wieder links oberhalb der Knoten angegeben, die Postorder rechts unterhalb.

In Abbildung 8 sind die Knoten des XML-Dokuments gemäß der (*Preorder*, *Postorder*)-Nummerierung in ein zweidimensionales Koordinatensystem eingetragen. Verbindet man die Knoten, so kann man leicht den ursprünglichen Baum (schräg nach links geneigt) wiedererkennen. Unterteilt man, ausgehend von Knoten *i* das Koordinatensystem in 4 Quadranten, so ergibt sich folgende Aufteilung:

- im linken oberen Quadranten befinden sich alle *ancestor*-Elemente,
- im rechten oberen Quadranten alle *following*-Elemente,

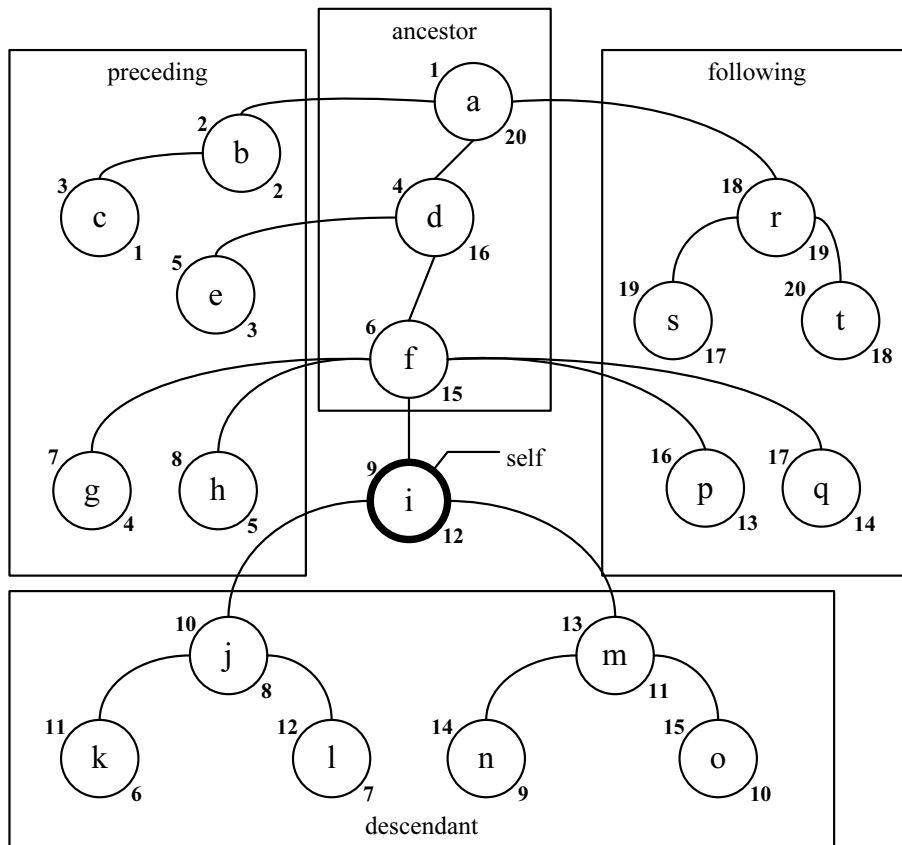


Abbildung 7: Benennung der Knoten und Nummerierung in Preorder (jeweils links oben) und Postorder (jeweils rechts unten)

- im linken unteren Quadranten alle *preceding*-Elemente und
- im rechten unteren Quadranten alle *descendant*-Elemente.

Literatur

- [O’Neil et al., 2004] O’Neil, P., O’Neil, E., Pal, S., Cseri, I., Schaller, G., and Westbury, N. (2004). Ordpaths: Insert-friendly xml node labels. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, pages 903–908, Paris, France.
- [Tatarinov et al., 2004] Tatarinov, I., Viglas, S., Beyer, K. S., Shanmugasundaram, J., Shekita, E. J., and Zhang, C. (2004). Storing and querying ordered XML using a relational database system. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, pages 204–215, Paris, France.

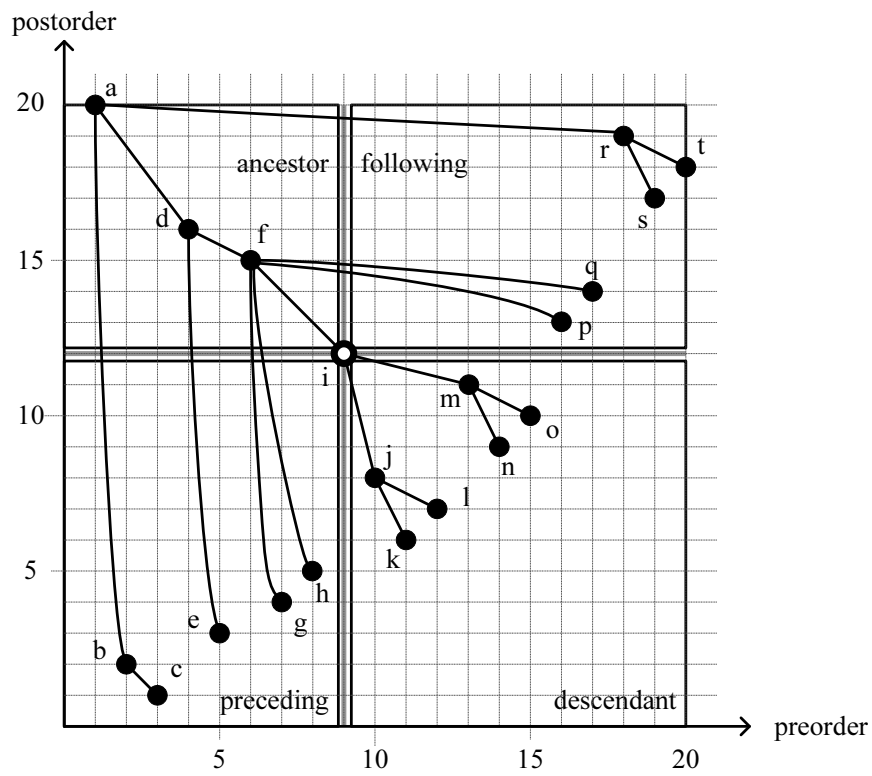


Abbildung 8: Eintragen der Knoten in ein zweidimensionales Koordinatensystem gemäß (Preorder, Postorder)