

# Bezugnahme auf Bestandteile von Gesetzestexten als Problem der angewandten Rechts- und Informationswissenschaften

Markus Lepper\* und Baltasar Trancón y Widemann\*

**Zusammenfassung** In zivilen freiheitlichen Gesellschaften müssen Gesetzestexte, Urteile, Kommentare, Ausführungsbestimmungen etc. für verschiedenste Anwendergruppen einfach zugänglich und inhaltlich verständlich sein. Dazu dienen mehr und mehr elektronische Medien, insbesondere digitaltechnische Aufarbeitungs- und Präsentationsfunktionen. Ein wichtiges Element sind dabei die in den Texten enthaltenen Bezugnahmen auf Bestandteile von Gesetzestexten. Eine genaue Analyse von deren Struktur und Bedeutung ist Voraussetzung für ihre informationstechnische Behandlung. Diese Analyse ist aber weit darüber hinaus auch erhellend für den alltäglichen Sprachgebrauch und die überraschend komplexen Eigengesetze der verwendeten Begrifflichkeit. Dieser Aufsatz klärt den Unterschied zwischen Informationsgehalt, Semantik und externer Repräsentation. Er schlägt vor eine einfache kleine formale Sprache, um die empirische Vielfalt letzterer exakter zu fassen und intellektuell wie technisch handhabbar zu machen.

## 1 Einleitung

Dieser Artikel analysiert Bezugnahmen auf Bestandteile von Gesetzestexten aus informationstheoretischer und rechtlicher Sicht. Die Ergebnisse sind Voraussetzung für digitaltechnische Implementierungen, bieten aber auch alltagstaugliche Erkenntnisse über deren mathematische, logische und sprachliche Struktur.

### 1.1 Die Zugänglichkeit zu Gesetzestexten als rechtliche und digital-technische Aufgabe

Gesetzestexte sind eine wesentliche Grundlage für das Funktionieren ziviler freiheitlicher Gesellschaften. Dies ist eine Grundannahme moderner Rechtsstaaten. Wenn man dieser Grundannahme zustimmt, dann folgt daraus unmittelbar die Forderung, dass Gesetzestexte a) zugänglich, und b) für die Anwender so verständlich wie möglich sein müssen. Wird eine dieser beiden Grundforderungen

---

\* Dr.-Ing. Markus Lepper und Dr. rer. nat. Baltasar Trancón y Widemann sind Informatiker und forschen und entwickeln zu den Themen Compilerbau, Semantik von Computersprachen und korrekter Software unter der gemeinsamen Firma <semantics/> GmbH, Berlin.

nur unzureichend erfüllt, ist die Funktionalität der Gesetze gefährdet und zu meist nur mit nicht unerheblichem Mehraufwand – wenn überhaupt - herzustellen.

In modernen Gesellschaften erfolgt der Zugang zu Gesetzen<sup>1</sup> mittlerweile im wesentlichen – und in einigen Ländern bereits ausschließlich<sup>2</sup> – über elektronische Medien, vor allem über das Internet. Dies gilt sowohl für die Nutzergruppe der professionellen Rechtsanwender wie für die der rechtssuchenden Bürgerinnen und Bürger. Weiterhin ist ein Erschließen der relevanten Gesetzestexte in Anbetracht a) ihrer Menge und b) ihrer unterschiedlichen Herkunftsorte (EU, Bund, Länder, Kommunen, andere) beiden Nutzergruppen fast nur noch mittels professionell aufbereiteter Gesetzesdatenbanken möglich.

Es ist deshalb eine naheliegende Forderung, dass die Anwender der methodischen Kriterien zur „Herstellung von verständlichen Gesetzestexten“ (Juristen) und derjenigen zur „informationstechnischen Verarbeitung von Gesetzestexten“ (Informatiker) fachübergreifend miteinander in den Dialog treten. Das Ziel ist dabei, zu schauen, welche Erkenntnisse sich wechselseitig aus den Anforderungen und guten Praxisstandards der jeweiligen Disziplinen für das eigene Arbeitsgebiet ergeben, und wie diese zu einem vertieften Verständnis der eigenen Arbeit beitragen könnte.

Obwohl es dazu eine ganze Reihe begrüßenswerter Ansätze gab und gibt<sup>3</sup>, ist diese Entwicklung noch durchaus am Anfang. Hier einen Beitrag zum besseren interdisziplinären Verständnis zu leisten, ist das Ziel dieses Artikels.

## 1.2 Ansatzpunkt: Intertextualität der Rechtsnormen

Ansatzpunkt ist die für Gesetzestexte spezifische hohe *Intertextualität* bei Erstellung und Gebrauch von Rechtsvorschriften. Diese hat a) eine syntaktische Dimension in der textlichen Darstellung, und b) eine – und je nach Zusammenhang auch mehrere – semantische Dimensionen für die zutreffende Erschließung (Gewinnung) des relevanten Inhalts der Rechtsnorm.

Gesetzestexte werden aus einer Mischung aus allgemeiner Umgangssprache und tradierter Fachsprache hergestellt. Sie sind damit notwendigerweise an die Genauigkeitsgrenzen dieser Sprachkörper gebunden – und damit zwangsläufig partiell ungenau.

Die avancierten Verfahren der digitalen Datenverarbeitung, wie maschinelles Lernen oder Computerlinguistik, werden immer besser darin, mit unscharfen Definitionen umzugehen und aus wenig formalisierten Eingaben befriedigende Ergebnisse zu liefern. Andererseits gibt es Teilbereiche der Fachsprache, deren

<sup>1</sup> hier gemeint im weiteren Sinne des materiellen Gesetzes, also aller geltenden Rechtsvorschriften

<sup>2</sup> Vgl. „Der gesamte Prozess der Gesetzgebung wird in Österreich vom Entwurf bis zur Kundmachung durch ein elektronisches Verfahren (E-Recht) unterstützt.“ (<https://www.parlament.gv.at/PERK/GES/ERECHT/>)

<sup>3</sup> Z.B. OASIS Legal Citation Markup (LegalCiteM) Technical Committee (<https://www.oasis-open.org/committees/legalcitem/charter.php>); MetaLex (<http://www.metalex.eu/>)

exakte Formalisierung aus prinzipiellen Gründen stets unabdingbare Voraussetzung für die automatisierte Behandlung bleiben wird, da sie bereits in ihrer traditionell-alltäglichen Gestalt rein formalistisch konstruiert sind. Dazu gehören z.B. Aktenzeichen, Autokennzeichen, geographische Koordinaten, Fundstellenangaben und die in diesem Artikel besprochenen Verweise auf Bestandteile von Gesetzestexten. Obwohl diese Bereiche auf den ersten Blick überaus trivial erscheinen, verglichen z.B. mit linguistischen Wortfeldanalysen der Texte selber, zeigt dieser Artikel im folgenden eine überraschende Fülle an Problematiken und Varianten.

Sobald eine informationstechnisch(-mathematische) Modellierung angestrebt wird, ist ein notwendiger Schritt, die Unschärfen und Usancen des historisch-geisteswissenschaftlich gewachsenen Symbolsystems „Gesetze“ aufzuweisen, zu analysieren, und in eine hinreichend exakte Form zu übersetzen.

Dies geschieht durch Anwendung von analytischen und modellierenden (=nachbildenden) Methoden der Informatik. Es entspricht aber aufgrund eben der historisch-tradierten Herkunft der Rechtsetzung durchaus dem Bedürfnis des Anwenders, weiterhin in einer „unscharfen“ oder „intuitiven“ Weise mit den Symbolen der Gesetzessprache operieren zu wollen.

Das im Titel angesprochene Problem besteht also nicht nur im Erstellen der informationstechnisch-mathematischen Definition, sondern weitergehend auch darin, sie in einer dem fachlich tradierten, üblichen Umgang kompatiblen Form zu treffen.

Wenngleich auch der Einsatz der digitalen informationsverarbeitenden Technologie der Auslöser ist, der die Lösung dieses Problems unumgänglich macht, so ist es doch von entscheidender Wichtigkeit, dass es sich dabei *nicht* um ein technologisches, sondern um ein logisches Problem handelt. Die Analyse und Synthese derartiger Symbolsysteme ist nicht beschränkt auf digitale Verarbeitung, sondern soll auch ganz unabhängig davon allen Beteiligten ein klareres Verständnis der Eigengesetze der Symbolsysteme vermitteln, mit denen sie täglich ganz selbstverständlich umgehen.

### 1.3 Kontext, Aufbau und Zweck des Artikels

Zum Thema der Gliederungsstruktur deutscher Gesetze fanden wir nur die Arbeit von Hamann.<sup>4</sup> Dieser lesenswerte Artikel scheint die erste systematische und umfassende empirische und theoretische Betrachtung dieses Themas zu sein. Er behandelt allerdings fast ausschließlich die Gliederungsebene *oberhalb* der Paragraphen, während es hier ja um Referenztechniken geht, die sich von dieser Ebene an nach unten erstrecken.

Im nächsten Abschnitt wird die Bezeichnung von Bestandteilen von Gesetzestexten als ein datentechnisches Problem erörtert. Es wird der eigentliche Informationsgehalt herausgearbeitet. Danach werden, auch zur Klärung der theoretischen Diskussionen im Allgemeinen, verschiedene Semantiken derselben syntaktischen Struktur aufgewiesen.

<sup>4</sup> Hanjo Hamann: Das Buch der Bücher, im Original mit Untertitel, Zeitschrift für Gesetzgebung, 2015, S.381ff.

In Abschnitt 3 wird dann die Trennung der internen Datenstruktur von ihrer externen Repräsentation erläutert. Für diese wird eine kleine Deklarationsprache vorgeschlagen. Abschnitt 4 beschreibt einige der unumgänglich notwendigen Rechenoperationen mit diesen Daten. In Abschnitt 5 werden sowohl konkrete als auch grundsätzliche Erkenntnisse aus der Analyse zusammengefasst.

Über das konkrete Thema hinaus versucht dieser Aufsatz, zentrale Begriffe der Informatik: Informationsgehalt, externe Repräsentation, Semantik, u.a., zu erläutern und für eine breite, grundsätzliche und mitnichten technologiebestimmte Diskussion fruchtbar zu machen.

## 2 Bezeichnungen von Bestandteilen von Gesetzen und ihre möglichen Semantiken

### 2.1 Rechtstatsächliches Vorgehen beim Verweis auf Gesetzestexte

Die Bezeichnung eines Bestandteiles eines Gesetzestextes<sup>5</sup> und damit einer gesetzlichen Regelung ist in vielen Kontexten an strenge rechtsförmliche Vorgaben gebunden.<sup>6</sup> Die herrschende Praxis besteht darin, nachdem das Gesetz als solches und die gemeinte kalendarische Fassung identifiziert ist, die Bezeichnung mit der *Nummer der Einzelvorschrift* fortzusetzen. Dabei ist Einzelvorschrift eine definierte Ebene der hierarchischen Gliederung, meist „Paragraph“ (ggfls. „Artikel“, deren Vertreter eindeutig als Textsegmente abgegrenzt und erkennbar sind. Es ist gewährleistet, dass diese Ebene durchlaufend nummeriert ist, also eine einfache Zahlenangabe zur eindeutigen Identifikation innerhalb des Gesetzes ausreicht.<sup>7</sup>

Sollen kürzere Textsegmente bezeichnet werden, so wird eine weitere hierarchische Verfeinerung herangezogen und die Positionnummer des jeweils zu bezeichnenden Segments auf den entsprechend tieferen Ebenen an die Gesamtkoordinate angehängt.<sup>8</sup>

Eine typische solche Hierarchie lautet z.B.:

- Ein Paragraph enthält direkt einen Satz oder mehrere Sätze, oder aber Absätze,
- Ein Absatz enthält einen Satz oder mehrere Sätze,
- Ein Satz enthält Text und/oder Listen,
- Eine Liste enthält Listenelemente,
- Ein Listenelement enthält Text und/oder Listen.

<sup>5</sup> Die im folgenden verwendete Nomenklatur richtet sich soweit wie möglich nach HdR = Handbuch der Rechtsförmlichkeit, Herausgegeben vom Bundesministerium der Justiz 3., neu bearbeitete Auflage 2008, online unter <http://hdr.bmj.de>, hier: Rn. 195.

<sup>6</sup> Vergleiche § 267 (3) StPO.

<sup>7</sup> Für Hanswerner Müller, Handbuch der Gesetzgebungstechnik, Köln 1963, betont diese Ebene ausdrücklich als „Einheit der Gliederung“ (S.41). Siehe auch HdR Rn.368 bis 370.

<sup>8</sup> HdR Rn.196 ff.

Die typischen Formulierungen solcher Bezeichnungen lauten dann „Paragraph 595 Absatz 1 Satz 1 Nummer 2 BGB“ oder „§ 595 (1) 1 Nummer 2 BGB“, wobei, entgegen der gemeinten Gliederungshierarchie, die Bezeichnung des Gesetzes als Ganzes meist nachgestellt wird.

## 2.2 Einbettung in ein allgemeines Textmodell

Sowohl die Gliederungsstruktur oberhalb wie auch die ab und unterhalb der Einzelschrift kann auf ein allgemeines Modell von Text abgebildet werden, siehe die (stark schematisierte) Abbildung 1.

Zunächst einmal kann der gesamte Gesetzestext als eine lineare Folge von Buchstaben aufgefasst werden.

Zum zweiten kann diese Folge aufgefasst werden als organisiert in Entsprechung zu einem *geordneten, gerichteten Baum von Textknoten*. Dies ist ein übliches Modell für Textobjekte im Allgemeinen, wie es z.B. Standardtechnologien wie SGML, HTML und XML zugrundeliegt: Jeder Knoten enthält eine geordnete Folge von zusammenhängenden Textsegmenten und/oder Kindknoten. Der *Gesamtinhalt* (deep content) eines Knotens ist die Aneinanderhängung der Textfragmente und der Gesamtinhalte der Kindknoten, in ihrer definierten Reihenfolge. Dabei ist entscheidend, dass diese der Reihenfolge der jeweiligen Textinhalte entspricht, und diese lückenlos und nicht-überlappend den gesamten Text abdecken.<sup>9</sup>

Im Sinne dieses Knotenmodelles sind die Paragraphen/Einzelschriften eine bestimmte Schicht von Knoten aus der Gesamthierarchie des Baumes. Darunter liegen dann die Knoten, die den Absätzen entsprechen, dann den Sätzen, dann den Listenelementen und Listenelementen, etc. Darüber liegen die von uns nicht betrachteten Titel, Kapitel, etc., die im semantischen Datenobjekt für unsere Zwecke auch weggelassen werden könnten (in der Abbildung gestrichelt), so dass die Paragraphen unmittelbare Kindknoten des Wurzelknotens wären. Dieser repräsentiert den Text als Ganzes.

Was ist unter solcher Modellbildung der konkrete Informationsgehalt eines Bezeichners? Es ergibt sich eine recht einfache Datenstruktur, dargestellt in Tabelle 1.

- Jede Bezeichnung einer Komponente eines Gesetzes erfolgt durch ein *geordnetes Tupel* (in Tabelle 1: *P* für „Position“). Dieses besteht aus folgenden Teilen:
- Zuerst die Identifikation des referierten Gesetzes als Ganzem, durch ein Element aus *L*.

<sup>9</sup> Siehe auch <https://wiki.selfhtml.org/wiki/XML/Regeln/Baumstruktur>

$L = \{\text{BGB}, \text{StGB}, \text{StVO}, \dots\}$  // = vorgegebene Identifizierer

$C = //$  Kalenderdatum der Form JJJJMMTT

$D = \mathbb{N}_0^+$

$P = L \times \text{opt } C \times D$

**Tabelle 1.** Datentypen der Stellenangaben

- Dann ein kalendarisches Datum aus  $C$ , welches den Zeitpunkt angibt, zu welchem die referierte Fassung des Gesetzes ausgefertigt worden ist.<sup>10</sup> Diese Angabe ist optional, kann also u.U. fehlen.<sup>11</sup>
- Dann folgt eine Folge von Koordinaten aus  $D$ . Jede Koordinate gibt die Position des gemeinten Knotens auf der entsprechenden Ebene an.
- Die Folge beginnt mit der explizit angegebenen Nummer der Einzelvorschrift.
- Die darüberliegenden Gliederungsebenen werden nicht benannt.
- Von den darunterliegenden Gliederungsebenen werden genau so viele (unmittelbar auf einander folgende) benannt, wie es zur präzisen Eingrenzung des gemeinten Bestandteils nötig ist. ( $D$  ist vom Datentyp  $\mathbb{N}_0^+$ , das sind alle möglichen nicht-leeren endlichen Listen aus natürlichen Zahlen, einschließlich der Null).

Hier ist bereits auffällig, dass der Informationsgehalt eine abstrakte mathematische Hintergrundstruktur bildet, die als solche nicht sichtbar wird. Davon unterscheidet die Informatik die *externe Repräsentation*, die den Informationsgehalt als schreib- und lesbaren Text codiert. Fast alle Vorschriften des Handbuchs der Rechtsförmlichkeit beziehen sich auf die externe Repräsentation. Wenn z.B. in den Formeln der Tabelle 1 die Menge  $L$  als  $\{\text{BGB}, \text{StGB}, \text{StVO}, \dots\}$  angegeben wird, so enthält diese rein symbolische Größen, die für bekannte, unterscheidbare und nur mit sich selbst identische Entitäten stehen. Dies sagt jedoch nichts aus über ihre lesbare textuelle Benennung, die, ganz anders geregelt, in verschiedenen Kontexten ja eine durchaus unterschiedliche sein muss.<sup>12</sup>

Ähnliches gilt für die externe Repräsentation von  $C$ : Die Zitervorschriften bestimmen genau, wie „Verkündigungsdatum“, „Fundstelle“ und „Kette aller Änderungen“ aufgeführt werden müssen.<sup>13</sup> Dies gehört aber mitnichten zum Informationsgehalt, sondern ist von der einfachen Nennung des Datums der Ausfertigung „funktional abhängig“: Sobald dieses Datum feststeht, gibt es, wegen der Faktizität der Geschehenen, grundsätzlich nur eine einzige korrekte Schreibweise, die zum Informationsgehalt folglich nichts beiträgt.

<sup>10</sup> HdR Rn.174

<sup>11</sup> Der Datentyp in Tabelle 1 ist entsprechend  $\text{opt } C$ . Dies erlaubt, den speziellen Wert  $\perp$  anzunehmen, der für „unspezifiziert“ steht. Ein Bezeichner mit dieser Komponente entspricht u.a. den „gleitenden Verweisen“, HdR Rn.243-245.

<sup>12</sup> Vgl. die Ausführungen zu den Begriffen Vollzitat, Zitiername, Kurzbezeichnung etc. im HdR Rn. 173 ff, 331 ff und 341 ff.

<sup>13</sup> HdR Rn. 174ff und 189ff.

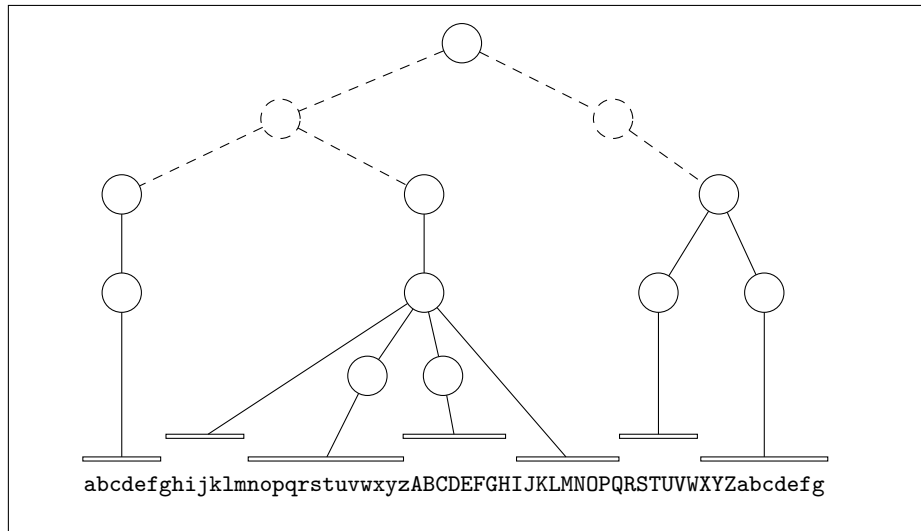


Abbildung 1. Beispiel für ein Datenobjekt als Ziel von Knoten- und Segmentsemantik

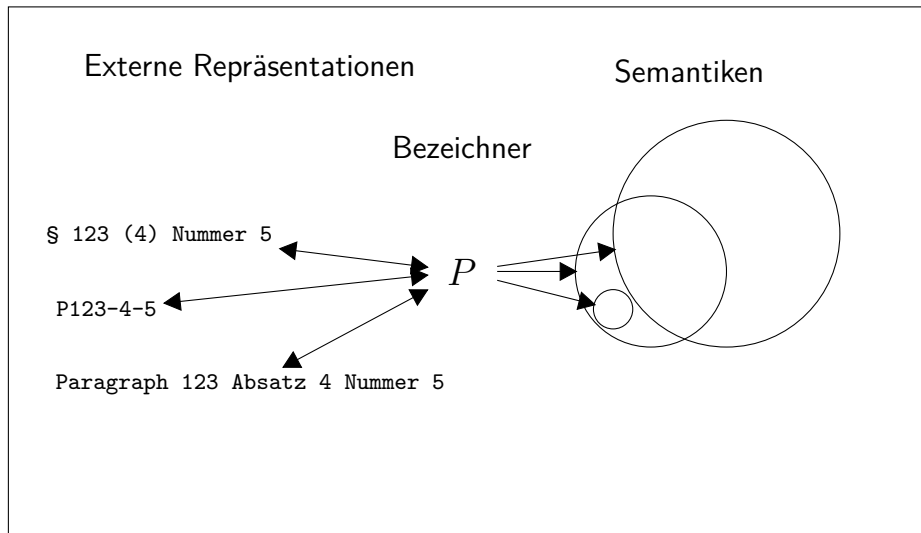


Abbildung 2. Verhältnis eines Bezeichners zu verschiedenen externen Repräsentationen und Semantiken („P123-4-5“ ist eine mögliche computer-interne Codierung, z.B. für HTML-Verweise)

### 2.3 Verschiedene Semantiken der Bezeichner aus $P$

Den verschiedenen Elementen einer syntaktischen Struktur wie  $P$  kann nun eine *Semantik* oder *Bedeutung* zugeordnet werden, siehe den rechten Teil von Abbildung 2. Dafür sind zumeist verschiedene Alternativen möglich, – im vorliegenden Falle ist es sogar so, dass verschiedene Semantiken gleichberechtigt nebeneinander stehen und in verschiedenen Situationen zur Anwendung kommen.

Die Definition solcher mathematischen Semantiken ist immer ein wesentlicher Schritt bei der Anwendung von Methoden der Informatik auf geisteswissenschaftliche Strukturen. Meist werden nämlich in der täglichen Praxis eh schon verschiedene Semantiken parallel verwendet und zwischen diesen gewechselt und vermittelt, aber ohne dass diese explizit unterschieden werden. Die explizite Modellierung ist also wiederum ein Beitrag zur „Klarwerdung des Geistes über sich selbst“, um es Hegel’sch auszudrücken.

„Semantik“ bedeutet in der Informatik, dass eine Funktion definiert wird, die den syntaktischen Strukturen (hier: den Elementen aus  $P$ ) Objekte aus einer *semantischen Sphäre* zuordnet. Diese enthält ebenfalls mathematische Objekte, die aber nun nicht mehr die Bezeichner, sondern das von diesem Gemeinte realisieren. Dafür sind fast immer, so auch hier, verschiedenartige Modellbildungen möglich, wie oben in Abschnitt 2.2 erläutert bezogen auf Abbildung 1.

Zunächst die *Knotensemantik*: Diese ordnet jedem Bezeichner einen Knoten der Text-Hierarchie zu, wie oben beschrieben. Dabei gibt die erste Zahl aus der Liste (aus  $D$ ) die explizite Nummer des Knotens auf der Ebene der Einzelschriften an, normalerweise ein Paragraphenknoten. Die jeweils nächste Zahl  $n$  in der Liste bewirkt dann immer, zum  $n$ -ten Kindknoten des erreichten Knotens fortzuschreiten, bis die Liste der Koordinaten (aus  $D$ ) erschöpft ist. Diese Semantik ist *partiell*: Es kann sein, dass einer Zahl in der Liste kein Kindknoten entspricht. Dies ist dann ein zwar syntaktisch korrekter Verweis, aber mit einer *leeren Semantik*.

Die *Segmentsemantik* ordnet danach jedem Bezeichner den von dem gefundenen Knoten bedeckten Gesamthalt zu, also einen zusammenhängenden Teilbereich der gesamten Buchstabenfolge.

Die *Punktsemantik* geht von diesem über auf dessen Anfangspunkt. Eine Präsentations-Software, z.B. eine verlinkte HTML-Seite, die der Benutzer anweist „zeige mir Paragraph 3 Absatz 4“, und die daraufhin an den Anfang des entsprechenden Textes springt, folgt dieser Punktsemantik. Hingegen eine Software, die den Textinhalt des gesamten Paragraphen 3, Absatz 4 auf den Drucker schickt, folgt der Segmentsemantik.

Oben erwähnte Verwirrung der Semantiken in der täglichen Praxis ist nun leicht verständlich, da sich die verschiedenen Semantiken teilweise nur in wenigen Aspekten oder Situationen unterscheiden. So ist die Punktsemantik von „Paragraph 5“ dieselbe wie die von „Paragraph 5 Absatz 1“ und von „Paragraph 5 Absatz 1 Satz 1“. Nur in ihrer Knoten- und Segmentsemantik unterscheiden sich diese drei Bezeichner.

Eine weitere wichtige Variante könnte mit *Ergänzungsemantik* bezeichnet werden: Gegeben sei ein Urteil, welches sich explizit auf „BGB § 595 Absatz



1 Satz 1 Nummer 2“ bezieht, siehe Abbildung 3. Dann liefern die Knoten- und Segmentsemantik tatsächlich nur die durch die rechte Klammer angedeutete Zeichenfolge. Das Urteil bezieht sich aber selbstverständlich nicht auf diese, sondern auf den gesamten grammatikalisch vollständigen Satz, der gebildet wird durch die Zeichen der Segmentsemantik, ergänzt um die vorangehende und die nachfolgende Zeichenfolge, – im Bild grau unterlegt. Diese Ergänzungssemantik wird im folgenden von uns nicht weiter besprochen; es ist aber wichtig sich klarzumachen, dass dies eine weitere, grundsätzlich verschiedene und praktisch höchlichst relevante Semantik desselben syntaktischen Datentyps  $P$  ist.

Einerseits ist es zwar noch durchaus möglich, eine unaufwendige Formalisierung anzugeben, wenn wir z.B. die der Ergänzungssemantik entsprechenden Bestandteile automatisch ausdrucken wollen: Drucke den Gesamtinhalt des nächsthöheren Knotens, der kein Listenknoten ist, unter Auslassung aller Geschwister- und Onkel-Listenknoten.<sup>14</sup> Dies entspricht ebenfalls der grauen Fläche in Abbildung 3.<sup>15</sup>

Andererseits aber bezieht sich erwähntes Urteil ja nicht nur auf diese, nun erweiterte, Buchstabenfolge, sondern auf den gesamten inhaltlichen Kontext, beginnend bei Sachdefinitionen in anderen Paragraphen bis hin zu externen Texten in anderen Urteilen und Kommentaren. Die entsprechenden Semantiken wären dann ganze *Datenbank-Anfrage* mit dem Bezeichner als Schlüsselwort, die man als *historische Semantik* bezeichnen könnte. Damit ist aber allemal die Grenze der zweckmäßigen Formalisierung überschritten, die mit der Ergänzungssemantik erreicht worden ist, – siehe auch weiter unten Abschnitt 5.3.

Darüberhinaus gibt es für manche Bestandteile von Gesetzestexten sogar kulturhistorische, soziologische, zeitgeschichtliche oder gar tagespolitische Verwendungskontexte, die wiederum als eigene Semantikbereiche zu betrachten sind, wenn sie auch die historische Semantik durchaus überlappen, – man denke an „Paragraph zweihundertachtzehn“ oder „Paragraph einhundertfünfundsiebzig“ in den Überschriften von Reportagen oder historischen Monographien.

### 3 Externe Repräsentation von Bezeichnern aus $P$

#### 3.1 Übersetzungsfunktion in eine externe Repräsentation

Die externen Repräsentationen sind, wie erwähnt, die für Ein- und Ausgabe notwendigen sichtbaren Text-Codierungen der internen mathematischen Größen, hier: der Bezeichner aus  $P$ . Die Übersetzung zwischen diesen ist gleichsam das Gegenstück zu ihrer Semantik, siehe den linken Teil von Abbildung 2.

Die Grammatik-Definition in Tabelle 2 zeigt eine kleine Sprache namens *Lft-Code* zur Beschreibung der möglichen externen Repräsentationen,<sup>16</sup> also der

<sup>14</sup> Analog zum Begriff „Kindknoten“ können weitere Verwandtschaftsbezeichnungen zur Navigation in einem gerichteten Baum verwendet werden.

<sup>15</sup> NB bedient sich diese Regel nur expliziter Syntaxelemente, nicht etwa Importe aus der Grammatik.

<sup>16</sup> Der Name ist aus der Bezeichnung der Java-Klasse übernommen und steht für `LocationFormat`.

§ 595 (1) <sup>1</sup> Der Pächter kann vom Verpächter die Fortsetzung des Pachtverhältnisses verlangen, wenn

1. bei einem Betriebspachtverhältnis der Betrieb seine wirtschaftliche Lebensgrundlage bildet,
2. bei einem Pachtverhältnis über ein Grundstück der Pächter auf dieses Grundstück zur Aufrechterhaltung seines Betriebes, der seine wirtschaftliche Lebensgrundlage bildet, angewiesen ist

und die vertragsgemäße Beendigung des Pachtverhältnisses für den Pächter oder seine Familie eine Härte bedeuten würde, die auch unter Würdigung der berechtigten Interessen des Verpächters nicht zu rechtfertigen ist.

<sup>2</sup> Die Fortsetzung kann ...

... des Bezeichners

(BGB, 20020102, (595, 0, 1, 0, 1, 2)) ∈ P

**Punktsemantik**

**Segment- und**

**Knotensemantik**

**Ergänzungssem.**

Abbildung 3. Verschiedene Semantiken desselben Bezeichners

sichtbaren schriftlichen Umsetzung der internen, rein mathematischen Daten. Die Syntax dieser Formalisierung folgt der seit über fünfzig Jahren bekannten und bewährten Methode des *printf()*-Befehles aus den Programmiersprachen C und Nachfolgern:

- Die in der Gesamtzeichenkette verstreut auftretenden Zeichen „%“ von links nach rechts stehen (grundsätzlich) für die verschiedenen Gliederungsebenen von oben nach unten.
- Sie eröffnen die *Konvertierungsgruppe* der jeweiligen Ebene, die mit dem eigentlichen *Konvertierungscode* abgeschlossen wird.
- Dazwischen stehen *Modifikatoren*.
- Diese Konvertierungsgruppe und der vor und nach ihr stehende Text zwischen den engsten umgebenden geschweiften Klammern wird nur dann gedruckt, wenn diese Gliederungsebene tatsächlich auftritt.
- Diese Klammern können enthalten konstanten Text, der unverändert aus- oder eingegeben wird, oder aber weitere Konvertierungsgruppen.
- Im konstanten Text wird mit eckigen Klammern die *Artbezeichnung* dieser Knotenebene<sup>17</sup> hervorgehoben. Diese wird mit „=“ eingeleitet, wenn sie in diesem Format zwar nicht gedruckt wird, aber zwecks Meta-Diskussion spezifiziert werden soll.

Im Einzelnen sind die Bedeutungen:

- Der Konvertierungscode gibt an, wie die „Zählbezeichnung“ dargestellt wird. Es bedeuten

<sup>17</sup> HdR Rn.379

- "1" – als arabische Zahl.
  - "i" – als kleine römische Zahl.
  - "I" – als große römische Zahl.
  - "a" – als kleiner lateinischer Buchstabe.
  - "A" – als großer lateinischer Buchstabe.
  - „α“ – als kleiner griechischer Buchstabe.
  - "L" – als Zahlwort der aktuellen Grundsprache (hier: Deutsch). Die Modifikatoren "m", "f" und "n" führen zu den Adjektivformen der Ordinalzahlen in den Genera maskulin, feminin, neutrum; ohne das erscheint das Zahlgrundwort/die Kardinalzahl.
- Die übrigen Modifikatoren bedeuten
- "0" – der Wert der Koordinate kann Null sein und wird dann nicht gedruckt.
  - "-" – Die Koordinate wird nicht gedruckt wenn sie eindeutig ist, wenn es also nur einen Kindknoten auf dieser Ebene gibt.
  - "~" – Die Koordinate wird nicht gedruckt wenn sie eindeutig ist, oder wenn die folgende Koordinate eindeutig ist, also nur einer der Geschwisterknoten überhaupt Kinder hat.<sup>18</sup>
  - ">" – Die darunterliegende Ebene wird durchnummeriert = ihr Zähler mit einem Wechsel auf dieser Ebene nicht zurückgesetzt.
  - "↑" – Der Druck erfolgt hochgestellt.
  - " \_ " – Der Druck erfolgt tiefgestellt.
  - "N" – Der Druck dieser Koordinate erfolgt auf einer eigenen Zeile und wird dadurch (zusätzlich) kenntlich gemacht.
- Die Zeichenfolge "%T" oder "%t" kann einmal irgendwo im umgebenden Text auftauchen und bedeutet dann, dass in den Daten der entsprechenden Gliederungsebene eine *Überschrift* enthalten sein muss oder kann, die an dieser Stelle des Textes gedruckt wird.
- Auch "%t" kann mit geschweiften Klammern und konstantem Text umgeben sein, der nur dann gedruckt wird, wenn tatsächlich eine Überschrift vorliegt.

Der Entwurf dieser Sprache ist durchaus praxisorientiert: Sie mischt munter Eingabe-, Ausgabe- und Meta-Formate und ist, sobald Varianten nebeneinandergestellt werden, nicht mehr redundanzfrei. Ein akademischer Entwurf sähe anders aus. Dennoch haben wir mit ihr gute praktische Erfahrungen gemacht.

### 3.2 Anwendung auf die Ebenen oberhalb der Einzelschrift

Die Gliederung des BGB oberhalb der Paragraphen<sup>19</sup> heißt im LftCode

```
{%nuL [Buch]. %T} {%nmL [Abschnitt]. %T} {%n-mL [Titel]. %t}
{[=Untertitel]%n-I. %T} {[=Kapitel]%n-1.{ %t.}} {[=Unterkapitel]%n->a){ %t.}}
```

<sup>18</sup> Diese und die vorangehende Regel sind notwendig für die irregulären Sonderfälle aus der Praxis bzgl. Satz und Liste, und werden unten in Abschnitten 3.3.2f. genauer besprochen.

<sup>19</sup> Die nicht sichtbaren Artbezeichnungen übernommen von Hamann, S.392.

```

lftCode ::= "{", fill, "%", modifier, convCode, fill, "}", lftCode?
fill    ::= ( text | "[" | "=" | text, "]" | "%I" | "%t" | lftCode ) *
text    ::= //beliebige Zeichenfolge, aber ohne "[", "{" und "%".
modifier ::= "0"? & ("-" | "~")? & ">"? & ("↑" | "_" | "N")? & ("m" | "f" | "n")?
convCode ::= "1" | "i" | "I" | "a" | "A" | "α" | "L"

```

**Tabelle 2.** Grammatik der Gliederungsbeschreibungssprache LftCode

Man beachte, dass die runden Klammern keine Sonderbedeutung haben, sondern normale gedruckte Zeichen sind.<sup>20</sup>

Dieser Code kann sowohl deskriptiv als auch normativ verwendet werden. So könnte ein Revisionsvorhaben eines so gegliederten Gesetzes beinhalten, die Ebenen von `{%nI. %T}{%n-1 %t}{%n->a} %t}` nach `{%nI. %T}{%n-1 %T}{%n->a}` aufzuräumen: Untertitel und Kapitel müssen danach Überschriften haben, Unterkapitel dürfen das nicht. Dies kann mittels des Codes konkret benannt und automatisiert überprüft werden.

Von den bei Hamann aufgewiesenen seltsamen Formulierungen nachträglicher Einfügungen<sup>21</sup> sind in diesem System nur die darstellbar, die auch in der Struktur des Bezeichners reine Einfügungen sind, nicht etwa Umstellungen, etc. Dies sind die dort aufgeführten letzten beiden:

„Dritter Abschnitt – Dritter Abschnitt a – Vierter Abschnitt“ ergibt sich aus der Deklaration „`{%nmL Abschnitt {%0a}}`“, ebenso „II. Abschnitt – III.1 Abschnitt – III. Abschnitt“ aus „`{%nI. {%01} Abschnitt}`“.

### 3.3 Ab und unterhalb der Einzelvorschriften

Für die Referenz in einen Gesetzestext (= Bezeichnung eines Gesetzesbestandteiles) sind die wichtigeren Gliederungsebenen, wie oben erwähnt, auf und unterhalb der *Einzelvorschriften*. Wichtiger Modifikator ist dabei "0": diese Gliederungsebene kann den Wert Null annehmen und wird dann im Druck weggelassen. Die *Erweiterungsbuchstaben* müssen nämlich von Anfang an als Gliederungsebene definiert werden. Deren Koordinate nimmt dann den Wert Null sogar als Normalfall an und wird erst in *späteren* Erweiterungen mit einem Wert ungleich Null in Erscheinung treten.

Nur wenn diese Ebene als eigene Knotenebene gleich von Anfang an modelliert wird, zu Beginn allerdings redundanterweise, wird die Knotensemantik aller internen Bezeichner von späteren Einfügungen nicht invalidiert werden können.

Im Falle des BGB lautet diese Spezifikation  
`{[=Paragraph] %1{%0a}.} {([=Absatz]%-1{%0a})} {[=Satz]%^1}`  
`{ [=Nummer]%1} { [=Buchstabe]%a}`  
 oder auch

<sup>20</sup> Die Modifikatoren ">" und "-" sind nur notwendig, wenn diese Spezifikation mit der darunterliegenden, ab „§“, konkateniert wird, was für unseren Zweck ja nicht geschieht.

<sup>21</sup> Hamann, S.394

{[Paragraph] %1{%0a}} { [Absatz] %-1{%0a}} { [Satz] %~1}  
 { [Nummer] %1} { [Buchstabe] %a}

Die interne Darstellung von „§ 594 (1) Satz 2“ ist also  $\langle 594, 0, 1, 0, 2 \rangle \in D$ , da die zukünftigen Einfügungen als Knotenebene schon vorgesehen werden müssen.

Im Gegensatz zu den verschiedenen Zitierformaten erscheint das wichtige Format P%1-%1-%1-%1-%1-%1-%1 niemals im menschenlesbaren Text. Vielmehr generiert es interne Bezeichner, die z.B. zur Codierung von HTML-Sprungzielen benötigt werden, der Form P123-0-4-0-1-2-3.

Dass für die Bezugnahme auf Gesetzesbestandteile nur diese unteren Ebenen relevant sind, ist von großem Vorteil für jede automatisierte Behandlung, ja, für jede auch nur um Systematik bemühte.<sup>22</sup> Allerdings erscheinen gerade auf dieser Ebene deutliche Unregelmäßigkeiten, die jeweils eigene Überlegungen notwendig machen:

**3.3.1 Der Satz als Gliederungsebene** Häufig sind Bezeichner bestehend aus den Einzelkoordinaten von Paragraph, Absatz und Satz, oder gar bis hinab zu Teil- oder Halbsatz. Damit entspricht „Satz“ der nächst-tiefere Knotenebene unterhalb von Absatz, in einem Modell wie in Abbildung 1. Während dieser aber noch mit typographischen oder orthographischen Mitteln explizit abgegrenzt wird, wird die Einheit des Satzes in den meisten Fällen aus der linguistischen/grammatikalischen Begriffsbildung übernommen.<sup>23</sup>

Dies bedeutet einen doppelten Paradigmenbruch: Einerseits wird aus einer anderen Sphäre, der von Grammatik und Zeichensetzung, ein Kriterium gleichsam importiert, während alle darunter und darüber liegenden Gliederungseinheiten explizit vom Autor eingetragen werden. Dies ist nicht unproblematisch, da z.B. automatische Erkennungen relativ aufwendig einen Abkürzungspunkt (oder heutzutage auch einen Trennungspunkt in einer URL) im Innern eines Satzes vom Satz-Ende-Zeichen unterscheiden müssen.<sup>24</sup>

Zum zweiten erfolgt die Identifikation nicht mehr durch explizit vergebene Nummern, wie auf allen Ebenen darüber, sondern nur durch die syntaktische Reihenfolge. Jeder spätere Einschub auf dieser Ebene invalidiert also sämtliche örtlich dahinter liegenden Koordinaten dieser Ebene in allen jemals zeitlich zuvor erzeugten Referenzen, – eine u.U. durchaus weitreichende Konsequenz. Bemerkenswerterweise wurde von privaten nachdruckenden Verlagen schon in alten Papierzeiten eine explizite Nummerierung der Sätze (durch hochgestellte Nummern, etc.) eingefügt, allerdings nur zu informatorischen Zwecken, ohne legalistische Wirkung.

<sup>22</sup> Der wild wuchernden Unsystematik der Ebenen darüber widmet sich ausführlich Hamann.

<sup>23</sup> Hw. Müller, S.43: „Die Einheit, also der Paragraph, kann aus mehreren, durch Punkte getrennten Sätzen, diese wieder aus durch Strichpunkt (Semikolon) getrennten Halbsätzen bestehen.“

Hamann, a.a.O., unterscheidet ausdrücklich „semantische“ und „juristische Gliederungseinheiten“, was wir der Sache nach teilen, jedoch anders benennen würden.

<sup>24</sup> Das Semikolon scheint da weniger problematisch.

Im Kontext des kleinen Entwicklungsprojektes der Verfasser wurde hingegen entschieden, auch eingabeseitig die Satznummern explizit zu vergeben und einzutragen. Im weiteren Verlauf wäre dann das Vorsehen von Erweiterungsbuchstaben, analog zu Paragraphen und Absätzen, nur konsequent.

**3.3.2 Alternative Ebenen unterhalb des Satzes** Eine wichtige Eigenschaft unseres semantischen Modelles (s.o. Abschnitt 2.2) ist die *Homogenität der Knotenebenen*. Sämtliche Knoten einer bestimmten Ebene tragen dieselbe *Artbezeichnung*<sup>25</sup>, und sie enthalten ausschließlich Kindknoten wiederum nur einer einzigen (anderen) Art. Dies wird nicht zuletzt dadurch erreicht, dass redundante Knoten eingefügt werden. Ein Paragraph, der direkt Text enthält, erhält im Knotenmodell dennoch einen einzigen Knoten der Absatz-Ebene, dessen Angabe redundant ist.

Dementsprechende Homogenität gilt folglich auch für die Koordinaten eines Bezeichners. Sie endet bei den in Praxis auftretenden Bezeichnern jedoch unterhalb des Absatzes. Dort sind als folgende Koordinate sowohl der Satz als auch der Listenpunkt üblich. Unterhalb des Satzes wiederum ist der Teil- oder Halbsatz üblich, oder auch ein Listenpunkt.

Selbstverständlich ist es möglich, Formen von Bäumen zu konstruieren, die heterogene Listen von Kindknoten erlauben. Oder auch Bezeichner, die auf einer bestimmten Ebene Alternativen von semantischen Bezugssystemen adressieren können. Diese Maßnahmen aber führten zu erheblichem Mehraufwand in allen Bereichen: der theoretischen Konstruktion und der praktischen Implementierung des Grundmodelles und aller zukünftig zu entwickelnden Verarbeitungsalgorithmen, in Dokumentation, Schulung und Verständnis.

Wir wählten deshalb einen anderen Ansatz:

- Die Koordinatenangabe des Teilsatzes wird nicht unterstützt. Wenn also eine Bezeichnung bis auf diese Ebene absteigt, dann ist die Folge von Koordinaten unterhalb des Satzes nicht mehr von der Formalisierung erfasst. Dies sollte kein grundsätzliches Problem bedeuten, das die Kombination formaler und nicht formalisierter Bestandteile eh immer notwendig sein muss, und die automatische Datenverarbeitung i.A. nicht auf diese Ebene hinabsteigt.<sup>26</sup>
- Die Satz-Nummer wird dann in der Praxis weggelassen, wenn ein Listenpunkt genannt wird, und nur einer von mehreren Sätzen überhaupt eine Liste aufweist. Das löst der LftCode durch den Modifikator „~“, der in diesem Fall die Unterdrückung der Ausgabe und Optionalisierung der Eingabe bewirkt.

<sup>25</sup> HdR Rn. 369 und 379.

<sup>26</sup> Im ganzen Bereich der automatisierten Linguistik wird selbstverständlich in noch viel feinere Bereiche hinabgestiegen. Dies erfordert aber per se ganz andere Verfahren und Betrachtungsweisen, mit denen verglichen die hier vorgestellten der expliziten juristischen Gliederung als sehr grobkörnig erscheinen. Nichtsdestotrotz können sich Überlegungen, Ergebnisse und Verfahren beider Ansätze sinnvoll ergänzen.

- Interessanterweise ist dieses Verfahren nicht üblich für Unterpunkte von Listenelementen (meist durch kleine Buchstaben angegeben). Beim Aufbau

```
§ 2
  1. a
    1. b
      2. a
        2. b
          2. c
```

wäre die Angabe „§ 2 Unterpunkt c“ eindeutig und mithilfe des lftCodes „`§1{ Nummer %-1}{ Unterpunkt %a}`“ zwar möglich, aber ungewöhnlich.

- Eine weitere Unregelmäßigkeit besteht darin, dass wir keinen Fall kennen mit *mehr als einer Liste per Satz*. Folglich muss eine auftretende Liste nicht als eigener Kind-Knoten des Satzes modelliert werden: es können direkt die Listenelemente dort eingehängt werden.

**3.3.3 Spätere Einschübe in Listen** Eine weitere Unregelmäßigkeit ist, dass die Unterpunkte von Listenelementen mit Buchstaben bezeichnet werden. Buchstaben stehen also nicht zur Kennzeichnung *später eingeschobener Listenelemente* zur Verfügung. Einschübe in die Mitte einer Liste sind deshalb nur durch Ummummerierung möglich und möglichst zu vermeiden, da durch sie bestehende Referenzen auf folgende Listenplätze ungültig -- oder noch schlimmer: verborgen irreführend -- werden.

Anfügungen an das Ende einer Liste sind hingegen, natürlicherweise, unproblematisch. Einen Einfügebuchstaben für Listenelemente vorzusehen hätte wohl auch ein sehr ungünstiges Kosten-Nutzen-Verhältnis: Immerhin verlängert es jede Bezeichnung um eine (zunächst redundante) Koordinaten-Ebene, die dann fast nie genutzt werden würde.

## 4 Strukturen über $P$ und Berechnungen darauf

### 4.1 Verschiedene Begriffe von „Bereich“

Endgültig unumgänglich wird die Trennung von externer Repräsentation und internem Datentyp  $P$ , sobald es an's *Rechnen* geht. Eine Hauptaufgabe des Digitalrechners bei der Präsentation von (fertigen oder entstehenden) Gesetzestexten ist das *Suchen*, und zwar nach komplexen, zusammengesetzten logischen Bedingungen. Egal ob die zugrundeliegenden Prädikate durch linguistische Analyse oder (wie im Projekt der Verfasser) durch explizites Markup zustandekommen, der Rechner soll immer Vereinigungen, Schnitte und Negation von *Text-Bereichen* berechnen können.

Wir menschlichen Benutzer meinen fast immer *inklusive Knotenbereiche*: „Sachverhalt XY wird behandelt in Paragraph drei bis Paragraph vier, Absatz zwei“ meint drei Knoten und ihren Textinhalt: den ganzen Paragraph drei und die ersten beiden Absätze von Paragraph vier.

Diese Semantik ist zum Rechnen sehr unangenehm: (a) Es gibt recht komplizierte Kürzungs- und Normalisierungsregeln. Z.B. „Paragraph zwei bis Paragraph zwei Absatz drei“ ist keine normalisierte Formulierung und damit kein sinnvolles Datum, da der zweite Knoten im ersten zur Gänze enthalten ist.

(b) Bei der Vereinigung von Bereichen muss *kontingente Information* berücksichtigt werden: „Paragraph zwei, Absatz zwei und Paragraph drei bis Paragraph sieben, Absatz zwei“ kann und muss (zwecks Weiterrechnen) zu „Paragraph zwei, Absatz zwei bis Paragraph sieben, Absatz zwei“ zusammengefasst werden, dann und nur dann wenn Paragraph zwei, Absatz zwei und Paragraph drei unmittelbar aneinanderstoßen.<sup>27</sup> Die Rechenregeln hängen also von der konkreten Gestalt des Gesetzes ab, was sie praktisch unwartbar macht.

Für alle automatischen Auswertungsprozesse ist zweckmäßig der Übergang auf *Punktbereiche*. Dabei stehen die Bezeichner aus  $P$  für die Punkte im Textfluss, wie oben beschrieben, und die Bereiche dazwischen sind das jeweils Gemeinte. „Paragraph eins, Absatz drei“ wird also z.B. durch ein Datum wie „(1,3 – 2)“ beschrieben, dann und nur dann wenn der Punkt am Ende von diesem Absatz zugleich der Anfang von Paragraph zwei ist, ansonsten durch „(1.3-1.4)“.

Der Übergang von den inklusiven Knotenbereichen auf die Punktbereiche ist es also allein, der ein für allemal die kontingente Strukturinformationen berücksichtigen muss. Alle weiteren Berechnungsregeln sind von ihnen unabhängig, und erst bei der Rückkonvertierung werden sie wieder benötigt.

#### 4.2 Berechnung von Referenztexten

Analoge Überlegungen gelten für die Berechnung verkürzter und relativer externer Repräsentationen von Referenzen: Ein und derselbe Knoten wird textuell als „Listepunkt b“ oder „Absatz 1, Listepunkt b“ oder „Paragraph 5, Absatz 1, Listepunkt b“ bezeichnet, je nach Standpunkt der Referenz: in derselben Liste, außerhalb dieser im selben Paragraphen, oder aber nur in demselben Gesetz.

Solche relativen Referenzen stecken meist auch in den externen Repräsentationen von Bereichen: Zweckmäßigerweise soll in der formal korrekten Bezeichnung „Paragraph eins Absatz eins bis [Paragraph eins Absatz] zwei“ bei der letzten Ausgabe an einen menschlichen Leser der eingeklammerte Teil ja weggelassen werden. Dies erfordert ähnliche Rechenoperationen.

### 5 Zusammenfassung und Auswertung

Wir haben im Vorangehenden ein Knotenmodell der Textstruktur aufgestellt, dann den Datentyp des Verweises  $P$ , indem der Informationsgehalt der praktisch üblichen Verweise herausdestilliert (und teilweise geglättet) wurde. Die-

<sup>27</sup> Das auch dann, wenn all diese Ergebnisse zuletzt dann doch wieder auseinandergenommen extern repräsentiert werden sollten, was sinnvoll scheint, sie also letztlich als „Paragraph zwei, Absatz zwei bis Ende, Paragraphen drei bis sechs und Paragraph sieben bis Absatz zwei“ ausgegeben werden. Davor aber muss dem Computer die faire Chance gegeben werden, nach überschaubaren Regeln zu rechnen.



sen Verweisen wurden verschiedene Semantiken zugeordnet (Knoten-, Bereichs-, Punkt-, Ergänzungssemantik, etc.).

Die Sprache *lftCode* beschreibt die andere Richtung der Übersetzung, nämlich verschiedene mögliche externe Repräsentationen für die Daten  $\in P$ .

Für die konkrete Arbeit und Weiterentwicklung lassen sich einige Hauptpunkte zusammenfassen:

### 5.1 Trennung von interner und externer Repräsentation

Ein wesentlicher Vorteil der Trennung von interner und externer Repräsentation ist die Austauschbarkeit letzterer: substantiell derselbe Bezeichner  $\in P$  kann in verschiedenen Formaten ein- und ausgegeben (gedruckt und parsiert) werden. So kann das Druckformat des Gesetzestextes vom Format der Zitation abweichen: Heißt es im BGB z.B. wie oben, kann ein Verweis darauf dem Format „Paragraph  $\%i\{\%0a\}\{$ , Absatz  $\%1\{\%0a\}\}$ “ haben, oder auch nur kurz „ $\%S\%i\{\%0a\}\{(\%1\{\%0a\})\}$ “.

Dies kann besonders in *multi-lingualen* und *trans-nationalen* Kontexten fruchtbar werden; *lftCode* unterstützt alle sinnvollen Formate beliebiger Rechssysteme.

### 5.2 Kosten

Alle Unregelmäßigkeiten verursachen hohe Kosten. So kann z.B. oben erwähnter Paradigmenbruch der aus der Grammatiktheorie „importierte“ Begriff des Satzes (s.o. Abschnitt 3.3.2) die Behandlung dieser einen Koordinaten-Ebene fast so „teuer“ machen wie die Erstellung des gesamten Restes des Systems.

„Kosten“ meint dabei nicht nur die aufzuwendende Arbeitskraft zur Erstellung von Theorie, technischem Plan und konkreter Software, sondern auch, vielleicht sogar vielmehr, den mentalen Aufwand zum Verstehen der Prinzipien und zum Erlernen des effektiven Umganges. Nicht zuletzt steigt durch solche Ausnahmen bedenklich die Gefahr, dass spätere Weiterentwicklungen und Anpassungen unbemerkt Unstimmigkeiten und Fehlerquellen in das existierende System tragen.

Es ist also im Zweifelsfalle immer eine orthogonale, mit dem Rest des Systems zusammenpassende Lösung zu bevorzugen, auch wenn dadurch lieb gewordene Eigenheiten aufgegeben werden müssen.

### 5.3 Formalisierung und Flexibilisierung haben vernünftige Grenzen

Andererseits hat aber auch die Formalisierung ihre Grenzen: Ab einem bestimmten Punkt der Detailliertheit sinkt das Kosten-Nutzen-Verhältnis dramatisch. (Beispiele sind an Ende von Abschnitt 2.3 der Übergang auf weitere Semantik-Begriffe jenseits der Ergänzungssemantik, und das Hinabsteigen in Teilsätze in Abschnitt 3.3.2.) Deswegen ist bei jedem Entwurf wichtig, auch weiterhin informelle Anteile mit den formalisierten auf einfache und eindeutige Weise kombinieren zu können.

Die vorgeschlagene Sprache LftCode dient ja gerade dazu, historisch eingeführte (leichte) Unregelmäßigkeiten auf eine äußerst regelmäßige Struktur (Listen von natürlichen Zahlen aus  $D = \mathbb{N}_0^+$ ) abzubilden und damit in einem neuen, stärker formalisierten Kontext weiterhin verwendbar zu halten. Es können also die Römischen Zahlen oder deutschsprachigen Zahlwörter als externe Repräsentation einfacher Zahlen verwendet werden.

Diese Flexibilisierung hat aber auch ihre Grenze: Für von Hamann aufgeführten stärker unregelmäßigen Einfüge-Konstruktionen wie „Fünfter Titel – 5a. Titel – Sechster Titel“ oder „Zweiter Abschnitt – Abschnitt IIa – Dritter Abschnitt“<sup>28</sup> ließe sich selbstverständlich auch eine Spezifikationsprache bauen. Der Aufwand läßt sich aber nicht vertreten, da dadurch ja nicht nur Eigenheiten, sondern gar Widersinnigkeiten gewürdigt und perpetuiert würden.

Die Abschätzung der Grenzen von Formalisierung und Flexibilisierung ist also immer Sache einer Einzelbeurteilung, auf einer inhaltlichen, sachlogischen Ebene. Die mathematischen Verfahren allerdings setzen die Grenzpföcke und damit wichtige Maßstäbe.

## Danksagung

Für wertvolle Hinweise und Beratung zu den legistischen Fragestellungen bedanken wir uns bei Herrn MinR Förster, Leiter der Normprüfstelle im Justizministerium Mecklenburg-Vorpommern.

---

<sup>28</sup> Hamann, S.394.