

Der Leser sollte sich beim flüchtigen Betrachten des folgenden Beitrags nicht durch die programmiersprachenähnlichen Passagen abschrecken lassen. Der Text bietet nämlich in klarer und verständlicher Form die Chance zu lernen, wie man eine kontextfreie formale Grammatik aufbaut (ein zentrales Thema der Informatik) und wie man diese praktisch nutzen kann. Das gewählte Beispiel gehört zum juristischen „Kernbereich“: Erkennen von „Gesetzeszitate“ (d. h. den Stellen, wo Vorschriften angeführt werden) in Gesetzestexten. Es verdient von daher Interesse und auch deswegen, weil es zeigt, wie ein Teilbereich bisher meist noch ausschließlich manuell betriebener Nachbearbeitung beispielsweise von Rechtsprechungsdokumenten EDV-gestützt möglich wäre. Die Beispiele sind dem Schweizer Recht entnommen. Da die Zitiergewohnheiten im deutschen Recht sich nicht ganz mit der Schweizer Praxis decken, verlangt eine Umsetzung des Systems in deutsche Verhältnisse geringfügige Modifikationen, die aber unschwer (gewissermaßen als Übungsaufgabe zum Beitrag) zu bewerkstelligen sind. (red.)

APLC: Ein Parser für Gesetzeszitate

Andreas Wendelspiess

Seit einiger Zeit werden immer mehr Texte in Volltextdatenbanken archiviert und von dort auch wieder abgerufen. Um solche Texte mit Hilfe von Stichwörtern oder Suchbegriffen suchen zu können, werden häufig Indextabellen generiert. Die meisten Textarchivierungssysteme können solche Indextabellen nur erzeugen, wenn die Suchbegriffe unstrukturiert sind (z. B. ein einzelnes Wort) und direkt im Text erwähnt werden. Der Autor beschreibt hier eine Lösung für speziell strukturierte Suchbegriffe – für Gesetzeszitate.

Problembeschreibung und Lösungsidee

APLC:
A Parser for Law Citations

In der Informatikforschungsgruppe der Schweizerischen Lebensversicherungs- und Rentenanstalt wurde ein System zur Erkennung von speziell strukturierten Suchbegriffen – von Gesetzeszitate – entwickelt, das im folgenden APLC (A Parser for Law Citations)-System genannt wird. „Gesetzeszitat“ steht hier für einen Verweis auf ein Gesetz und nicht für einen Auszug aus einem Gesetz.

Grundsätzlich besteht ein Gesetzeszitat aus dem Namen eines Gesetzes und der Referenz auf dessen Inhalt. Entsprechend der Struktur dieses Inhalts ist die Referenz eine Folge von arabischen und römischen Zahlen und Buchstaben und gibt eine Hierarchie der verschiedenen Einheiten wieder. Die hier zugrunde gelegte Hierarchie entspricht den schweizerischen (eidgenössischen) Gesetzeswerken.

Kurze Einführung in kontextfreie formale Sprachen

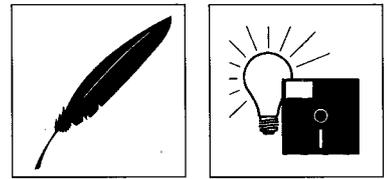
Für das APLC-System wurde eine kontextfreie formale Sprache definiert. Eine formale Sprache ist eine künstliche Sprache mit einem endlichen Symbolvorrat und einer fest definierten Grammatik. Eine kontextfreie Sprache ist eine formale Sprache mit einer kontextfreien Grammatik. Eine kontextfreie Grammatik ist definiert durch (vgl. [Aho]):

1. eine Menge von Terminal-Symbolen
z. B.: A, L
2. eine Menge von Nonterminal-Symbolen
z. B.: wort, satz
3. eine Menge von Grammatik-Regeln (Produktionen)
z. B.: wort -> LA; satz -> wort | wort satz

Die linke Seite jeder Regel besteht aus genau einem (nur bei kontextfreien Grammatiken!) Nonterminal-Symbol, die rechte Seite aus einer Folge von Terminal- und Nonterminal-Symbolen. Das spezielle Zeichen „|“ ist nicht Element der Sprache, sondern ist als Alternative zu verstehen und bedeutet die Zusammenfassung-zweier Regeln mit gleicher linker Seite. (Im Beispiel: Ein *satz* besteht aus einem *wort* oder aus einem *wort* gefolgt von einem *satz*.)

4. ein Startsymbol aus der Menge der Nonterminal-Symbole
z. B.: satz

Dipl. Informatik-Ing. ETH Andreas Wendelspiess (Zürich) war bis Ende Mai 1992 in der Forschungsgruppe der Schweizerischen Rentenanstalt tätig.



Symbolfolgen und Worte der Sprache

Mit den Terminal-Symbolen können Symbol-Folgen gebildet werden. Eine Folge von Terminal-Symbolen ist ein Wort der Sprache, wenn aus dieser Folge durch die umgekehrte Anwendung von einer oder mehreren Regeln das Startsymbol erzeugt werden kann oder umgekehrt, wenn die Folge durch Anwendung der Regeln aus dem Startsymbol abgeleitet werden kann.

Mit der obigen Grammatik können folgende Zeichenfolgen als Wörter bzw. Nicht-Wörter der Beispiel-Sprache erkannt werden:

Wörter: LA
LALALALA

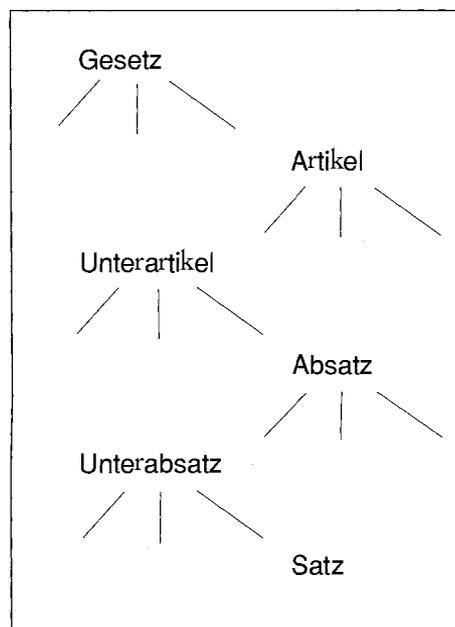
Nicht-Wörter: LLLLL
AAAA
LALL

Ein Gesetz ist unterteilt in Artikel oder Paragraphen. Artikel können weiter unterteilt werden in Unterartikel (werden zitiert mit „Ziff.“ oder „Buchst.“), die weiter unterteilt werden in Absätze (oder Alineas). Absätze können unterteilt werden in Unterabsätze (werden zitiert mit „Ziff.“, „Buchst.“ oder „lit.“) und weiter in Sätze (Abb. 1). Die Bezeichnungen „Unterartikel“ und „Unterabsatz“ wurden hier zum besseren Verständnis neu eingeführt. Nur selten werden in einem Zitat alle diese Stufen bezeichnet. In vielen Gesetzen werden auch nicht alle diese möglichen Hierarchiestufen zur Unterteilung gebraucht; sei es, daß nicht so tief strukturiert wird oder daß Zwischenstufen wie „Unterartikel“ fehlen.

Beispiel: OR Art. 327a Abs. 1
Art. 34 VVG Abs. 1 h
462 Ziff. 1 ZGB

Wie aus den Beispielen ersichtlich, kann der Gesetzesname vor, innerhalb oder nach der Referenz stehen. In [Oft] wird die erste Version empfohlen.

Da Gesetze auf so unterschiedliche Art zitiert werden, entstand die Idee, einen Parser für deren Erkennung einzusetzen. Dazu mußte eine formale Sprache definiert werden, die beschreibt, wie Gesetzeszitate in Texten auftreten können.



Gesetzesunterteilungen

*Abb. 1:
Hierachische
Struktur eines
Gesetzes*

Funktionsbeschreibung

Damit aus einem Text Gesetzeszitate erkannt werden können, wird anhand eines speziellen Wörerverzeichnisses, der Lex-Spezifikation, für jedes Wort geprüft, ob es Bestandteil eines Zitates sein könnte. Darauf wird dem Wort das zugehörige Symbol zugeordnet. Die so entstehende Symbolfolge kann man als einen Text in APLC-Sprache bezeichnen. Anhand der Syntax dieser Sprache können nun die Gesetzeszitate identifiziert werden.

Die Hauptfunktion des APLC-Systems – Erkennen von Gesetzesziten aus einem Text – kann so in zwei Teilfunktionen aufgliedert werden:

1. Scanner: Eingabeverarbeitung und lexikalische Analyse (Übersetzung des Eingabetextes in eine Folge von Symbolen der APLC-Sprache)
2. Parser: Parsen (Identifikation der Zitate im APLC-Text) und Ausgabe-Erzeugung

Scanner und Parser arbeiten folgendermaßen zusammen (Abb. 2, s. Folgeseite):

- Der Scanner arbeitet den Eingabetext bei jedem Aufruf immer nur so weit ab, bis er ein APLC-Symbol erkannt hat, und übergibt dieses dem aufrufenden Parser zusammen mit dem entsprechenden Text (Beispiel: Das Symbol DN kann z. B. den Wert 55 haben).

'Scannen' und 'Parsen'

Kooperation zwischen 'Scanner' und 'Parser'

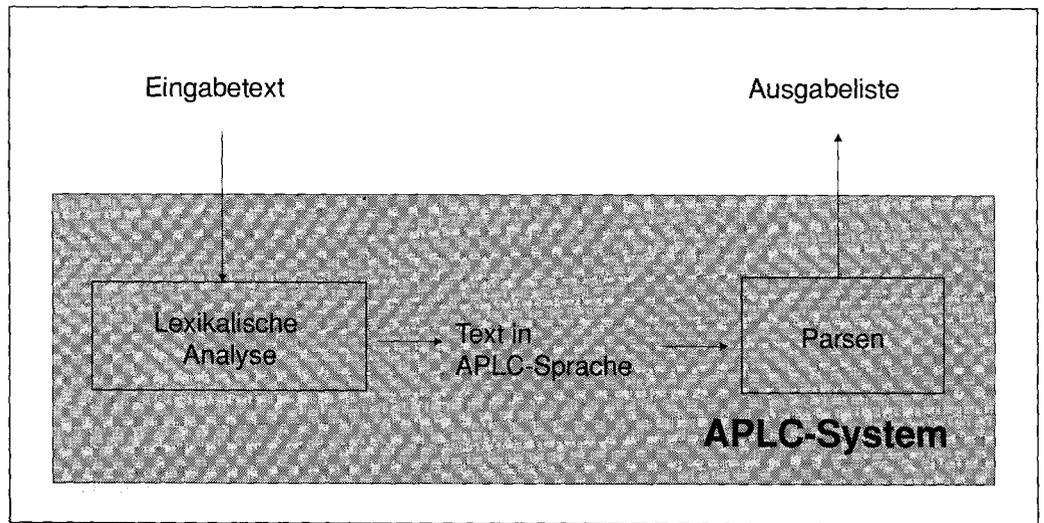
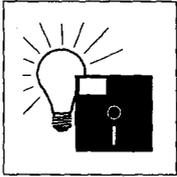


Abb. 2:
Die APLC-
Systemfunktionen

- Bis der ganze Eingabetext verarbeitet ist, ruft der Parser wiederholt den Scanner auf, um jeweils ein Symbol zu erhalten, und überprüft nach der APLC-Grammatik, ob dieses Symbol auf die bereits gelesenen folgen darf:
 - a. Falls ja, wird das Symbol zu dieser Folge hinzugefügt. Wird mit dem Symbol ein Gesetzeszitat beendet, wird dieses wohlformatiert in die Ausgabeliste eingetragen.
 - b. Falls nein, wird zwar der Eingabetext weiterverarbeitet, wie wenn kein Fehler aufgetreten wäre, die Folge der bereits gelesenen Symbole wird jedoch gelöscht, d. h., es wird simuliert, der Eingabetext begänne erst an der aktuellen Stelle.

Die Lex-Spezifikation

Die Lex-Spezifikation wird gebraucht, um Eingabetexte vom Deutschen in die APLC-Sprache zu übersetzen. Sie definiert die Abbildung von Wörtern im natürlichsprachlichen Text auf Terminal-Symbole der APLC-Sprache (Abb. 3).

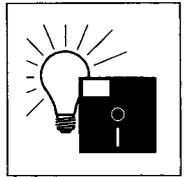
"Abs." "Absatz" "Al." "Alinea"	-->	AL
"Art." "Artikel"	-->	ART
"bis" "-"	-->	BIS
"und" ","	-->	CONJ
("0" .. "9")+	-->	DN
Name eines Gesetzes	-->	LAW
"a" .. "z" "A" .. "Z"	-->	LET
"Buchst." "Buchstabe" "Lit." "Iit."	-->	LIT
"bis" "ter" "quater" "a" .. "z" "A" .. "Z"	-->	NEXT
"I" .. "XIII"	-->	RN
"Satz" "Saetze"	-->	SATZ
"Ziff." "Ziffer"	-->	ZIF
("a" .. "z" "A" .. "Z")+ Spezialzeichen	-->	GARBAGE

Abb. 3:
Die Lex-Spezifi-
kationen

Dabei steht das Zeichen „|“ für Alternative, die Klammern „()“ bedeuten Zusammenfassung, die beiden Punkte „..“ stehen für „von ... bis“, „+“ für Repetition (1 oder mehrere Male), „Spezialzeichen“ für Interpunktion, Klammern, Prozent, Dollar, Paragraph ... und „Name des Gesetzes“ für die Spezifikation eines Namens wie im Abschnitt „Funktionsumfang“ beschrieben.

Die Buchstabenfolge „bis“, „ter“ usw. werden nur als NEXT interpretiert, wenn sie unmittelbar nach einer Dezimalzahl und vor einer Interpunktion oder vor einem Leerzeichen gelesen werden.

Ein Wort aus dem natürlichsprachlichen Text wird auf GARBAGE abgebildet, wenn keine andere Abbildungsregel zutrifft.



Die APLC-Sprache

Die APLC-Sprache ist eine kontextfreie formale Sprache, die durch die folgende Grammatik definiert ist:

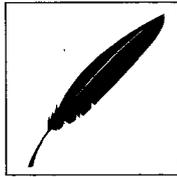
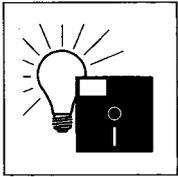
1. Die Terminal-Symbole

Dies sind die in der Lex-Spezifikation festgelegten Symbole:

AL, ART, BIS, CONJ, DN, GARBAGE, LAW, LIT, NEXT, SATZ, RN, ZIF.

2. Die Nonterminal-Symbole

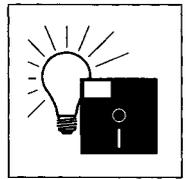
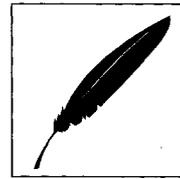
alinea	Absatz Bsp.: Abs. 5
alinea-list	Liste von Absätzen oder Bereichen von Absätzen Bsp.: Abs. 5, 7-9
alinea-part	Absatz, evtl. mit einer nachgestellten Liste von Unterabsätzen oder von Sätzen Bsp.: Abs. 5 lit. a, b, d-f
article	Artikel evtl. mit Nummerzusatz Bsp.: Art. 60 bis
article-list	Liste von Artikeln oder Bereichen von Artikeln Bsp.: Art. 108-110, 114
article-part	Artikel evtl. mit einer angehängten Liste von Unterartikeln oder von Absätzen Bsp.: Art. 267 Ziff. 1
citation	Gesetzeszitat Bsp.: Art. 34 VVG Abs. 1 h
citation-text	Text mit Zitaten Bsp.: Ich zitiere Art. 34 VVG Abs. 1 h zu Demonstrationszwecken.
drl	Dezimalzahl, Römische Zahl oder Buchstabe
law	Gesetzesname, evtl. im Genitiv, evtl. mit vorangestelltem „der“ oder „des“
satz-list	Liste von Sätzen oder Bereichen von Sätzen Bsp.: Satz 1, Satz 2
satz-part	Satz Bsp.: Satz 2
sub-al	Unterabsatz Bsp.: lit. a
sub-al-list	Liste von Unterabsätzen oder Bereichen von Unterabsätzen Bsp.: lit. a, b-d, f
sub-al-part	Unterabsatz, evtl. gefolgt von einer Liste von Sätzen.
subpar-list	Liste von Unterartikeln oder Bereichen von Unterartikeln Bsp.: Ziff. 1, Ziff. 2 bis Ziff. 5
subpar-part	Unterartikel evtl. gefolgt von einer Liste von Absätzen. Bsp.: Ziff. 1 AI. IV



3. Die Regeln

(Die Beispiele entsprechen den bereits für die Nonterminal-Symbole gegebenen.)

alinea	AL drl drl AL drl NEXT drl NEXT Ein Beispiel für eine Symbolfolge, die nach dieser Regel als <i>alinea</i> erkannt wird: AL DN
alinea-list	alinea-part alinea-part BIS alinea-part alinea-list CONJ alinea-part alinea-list CONJ alinea-part BIS alinea-part Ein Beispiel für eine Symbolfolge, die nach dieser Regel als <i>alinea-list</i> erkannt wird: AL DN CONJ DN BIS DN
alinea-part	alinea sub-al-list alinea satz-list alinea Ein Beispiel für eine Symbolfolge: AL DN LIT LET CONJ LET CONJ LET BIS LET
article	ART DN DN ART DN NEXT DN NEXT Bsp.: ART DN NEXT
article-list	article-part article-part BIS article-part article-list CONJ article-part article-list CONJ article-part BIS article-part Bsp.: ART DN BIS DN CONJ DN
article-part	article subpar-list article alinea-list article Bsp.: ART DN ZIF DN
citation	law article-list article-list law article-list law subpar-list article law alinea-list Bsp.: ART DN LAW AL DN LET
citation-text	{empty} citation-text garbage citation-text citation Bsp.: GARBAGE GARBAGE ART DN LAW AL DN LET GARBAGE GARBAGE GARBAGE
drl	DN RN LET
law	LAW DES LAW
satz-list	satz-part satz-part BIS satz-part satz-list CONJ satz-part satz-list CONJ satz-part BIS satz-part Bsp.: SATZ DN CONJ SATZ DN
satz-part	SATZ DN Bsp.: SATZ DN
sub-al	LIT LET LET ZIF DN Bsp.: LIT LET



sub-al-list	sub-al-part sub-al-part BIS sub-al-part sub-al-list CONJ sub-al-list sub-al-list CONJ sub-al-part BIS sub-al-part Bsp.: LIT LET CONJ LET BIS LET CONJ LET
sub-al-part	sub-al satz-list sub-al
subpar-list	subpar-part subpar-part BIS subpar-part subpar-part CONJ subpar-part subpar-part CONJ subpar-part BIS subpar-part Bsp.: ZIF DN CONJ ZIF DN BIS ZIF DN
subpar-part	ZIF DN alineal-list ZIF DN LIT LET alineal-list LIT LET Bsp.: ZIF DN AL RN

4. Das Startsymbol: citation-text

Mit der dargestellten Grammatik lassen sich Sätze formulieren wie:

1. GARBAGE GARBAGE LAW DN BIS DN GARBAGE GARBAGE (entsteht zum Beispiel aus: „wie in AHVG 53-66 beschrieben.“)
2. LAW ART DN CONJ DN NEXT BIS DN NEXT (entsteht zum Beispiel aus: „IVV Artikel 74, 74bis-74quater“)

Beispielsätze

Funktionsumfang

Durch die oben beschriebene Definition der APLC-Sprache und der Lex-Spezifikation ist es möglich, innerhalb gewisser Grenzen eine große Vielfalt verschiedener Zitierweisen abzudecken. So können Gesetzesnamen – wie im folgenden beschrieben – in verschiedener Gestalt erkannt werden.

*Bearbeitbar:
Eine Vielzahl von Zitierweisen*

1. Gesetzesnamen

Es gibt grundsätzlich drei verschiedene Arten, den Namen eines Gesetzes zu zitieren:

- a. die Abkürzung (z. B.: LG),
- b. den vollen Namen mit oder ohne Datum (z. B.: BG vom 8. Juni 1923 betreffend die Lotterien und die gewerbsmäßigen Wetten) und
- c. einen Kurznamen (z. B.: Lotteriegesezt).

Diese drei Möglichkeiten sind bezüglich Formulierung und Wortwahl in der Lex-Spezifikation genau beschrieben; es gibt wenige Abweichungsmöglichkeiten. So ist es erlaubt, folgende Abkürzungen auszusprechen:

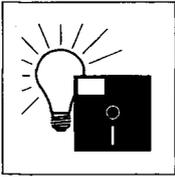
- BB -> Bundesbeschluss
- BG -> Bundesgesetz
- BRB -> Bundesratsbeschluss
- EG -> Einführungsgesetz
- V -> Verordnung
- VV -> Vollzugsverordnung oder Vollziehungsverordnung

Nur die ausgeschriebene Form wird dekliniert (nicht: „des BBes“, vgl. [Of]). Nicht erlaubt ist das Ausschreiben von Abkürzungen von Gesetzesnamen, wenn jene in Namen verwendet werden. So heißt es „VV zum LG“ und nicht „VV zum BG betreffend die Lotterien und die gewerbsmäßigen Wetten“.

Das APLC-System läßt zu, daß das Datum eines Gesetzes auch weggelassen werden kann. Wenn es jedoch erwähnt wird, müssen folgende Formatrichtlinien beachtet werden:

- a. der Tag wird mit einer oder zwei Ziffern gefolgt von einem Punkt angegeben,
- b. der Monat wird ausgeschrieben,
- c. das Jahr wird mit einer Zahl zwischen 00 und 99 oder zwischen 1000 und 2999 angegeben,
- d. Tag, Monat und Jahr sind mit einem Leerschlag getrennt,

*Gesetzesdatum:
Schreibweisen*



- e. das Datum kann mit einem Komma (davor oder danach) vom Namen abgetrennt sein,
f. vor dem Datum, aber nach einem eventuellen Komma steht „vom“.

2. Umlaute

Umlaute werden nicht erkannt. Das APLC-System erkennt nur Zeichen des ASCII-Zeichensatzes. Der Eingabetext muß also zuerst umgesetzt werden.

Beispiel:

BueG statt *BüG – BG ueber Erwerb und Verlust des Schweizerbuergerrechts*

3. Listen

Es ist möglich, in einem Zitat die ganze Hierarchie von Artikel, Unterartikel, Absatz, Literal hinabzusteigen, aber nicht wieder herauf.

Beispiel:

„*AHVV Art. 68 Abs. 1, AHVV Art. 69 Abs. 1*“ aber nicht „*AHVV Art. 68 Abs. 1, Art. 69 Abs. 1*“.

Entwicklungsumgebung

Unter UNIX in C

APLC wurde auf einer SUN-Workstation unter UNIX entwickelt, kann aber auch auf UNIX-PCs eingesetzt werden. Das APLC-System ist in der Programmiersprache C geschrieben und enthält keine hardware-spezifischen Programmteile; es sollte daher auch auf einem MSDOS-PC zu installieren sein.

Folgende Software-Hilfsmittel wurden benutzt:

- lex* 1. *lex* (*lexical analysis program generator*), um den Scanner zu erzeugen,
yacc 2. *yacc* (*yet another compiler-compiler: parsing program generator*) um die für die Erkennung der Zitate relevanten Komponenten des Parsers zu generieren.

Diese Software-Werkzeuge sind auf jedem UNIX-System verfügbar. *lex* übersetzt die Lex-Spezifikation in ein C-Programm (hauptsächlich bestehend aus der Prozedur *yylex()*, die von einem C-Programm aufgerufen wird, das mit *yacc* generiert wurde). *yacc* übersetzt ein Spezifikations-File, das die APLC-Sprache enthält, in ein C-Programm, das dann mit einem normalen C-Compiler übersetzt werden kann.

APLC und Text-Retrieval

Das APLC-System läßt sich mit einem Text-Retrieval-System verknüpfen (Abb. 4). Die Ausgabeliste kann dabei in eine Indextabelle integriert werden, so daß der Eingabetext später anhand eines gesuchten Zitates wieder aufgefunden werden kann.

Die anschließende Illustration soll dies verdeutlichen. Dabei wurde Artikel 66 des Bundesgesetzes über die Alters- und Hinterlassenenversicherung, Stand 1. Januar 1990, allein deswegen als Beispiel ausgewählt,

weil darin ein zweites Gesetz zitiert wird:

„Personen, welche als Organ einer Ausgleichskasse, einer Revisions- oder Kontrollstelle handeln oder welche sonstige Funktionen in einer Ausgleichskasse ausüben, haben die gleiche strafrechtliche Verantwortlichkeit wie Behördenmitglieder und Beamte gemäss den Artikeln 312-317 und 320 des Schweizerischen Strafgesetzbuches.“

Gefunden werden mit APLC die folgenden Zitate:

- AHVG Art. 66
- StGB Art. 312 – Art. 317
- StGB Art. 320

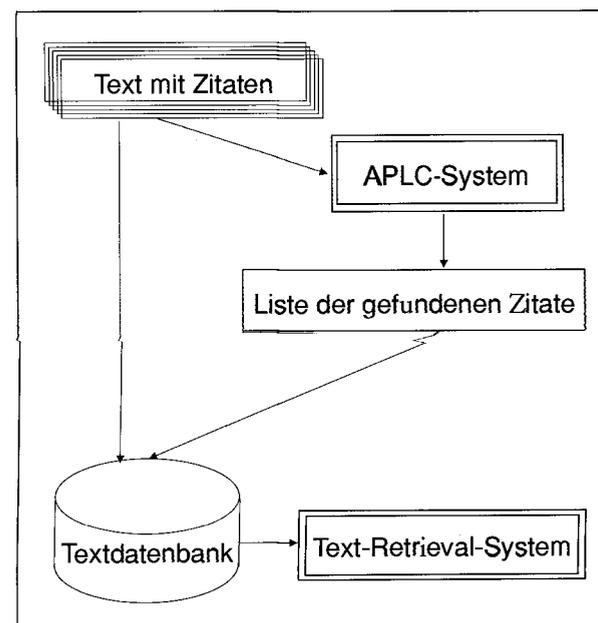
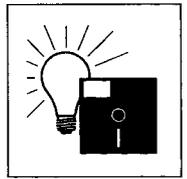


Abb. 4:
Einbettung des
APLC-Systems



Nachwort

Mit dem APLC-System konnte gezeigt werden, wie man aus einem Text Begriffe automatisch erkennen kann, auch wenn diese eine nur begrenzt einheitliche Struktur aufweisen. Das ist das wesentliche Resultat dieser Arbeit. Je nach Einsatzgebiet müßte die formale Sprache erweitert, geändert oder ganz neu definiert und/oder andere Werkzeuge (Parser- und Scanner-Generator) eingesetzt werden.

Die vorliegende Arbeit entstand in Zusammenarbeit mit Herrn Dr. M. Schieutermann, der auf den Bedarf an einem Erkennungsprogramm für Gesetzeszitate hingewiesen und das der APLC-Sprache und der Lex-Spezifikation zugrundeliegende juristische Wissen beigesteuert hat. Herr Dr. U. Reimer hat mit Anregungen und Korrekturen das Schreiben der Dokumentation und auch dieses Artikels hilfreich unterstützt.

Bibliographie

- Oft* Karl Oftinger
Vom Handwerkszeug des Juristen und von seiner Schriftstellerei
Zürich: Schulthess Polygraphischer Verlag, 1986
- Aho* Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman
Compilerbau
Bonn u.a.: Addison-Wesley, 1988

Bezugsquelle

Bei Einsendung einer 3,5"-Diskette erhalten Interessenten die beschriebene Software von:
Schweizerische Lebensversicherungs- und Rentenanstalt
Informatik-Forschungsgruppe
Dr. U. Reimer
Postfach
CH-8022 Zürich

jurpc.zip – jurpc.zip – jurpc.zip – jurpc.zip – jurpc.zip – jurpc.zip

Kündigung eines Leasingvertrages ohne Aufforderung zur vertragsgemäßen Gebrauchsüberlassung

BGH, Urteil vom 7. Oktober 1992 (VIII ZR 182/91)

Leitsatz

Zum Kündigungsrecht des Leasingnehmers aus BGB § 542.

Leitsätze der Redaktion

1. Nach § 542 BGB, der – soweit nicht wirksam etwas anderes vereinbart worden ist – grundsätzlich auch in dem vorrangig nach Mietrecht zu beurteilenden Leasingverhältnis anwendbar ist, ist der Leasingnehmer befugt, das Leasingverhältnis fristlos zu kündigen, wenn ihm der vertragsmäßige Gebrauch der Leasingssache ganz oder zum Teil nicht rechtzeitig gewährt wird und entweder der Leasinggeber eine ihm gesetzte Abhilfefrist ergebnislos hat verstreichen lassen oder die Bestimmung einer solchen Frist wegen Wegfalls des Erfüllungsinteresses entbehrlich war.
2. Die Vorenthaltung des vertragsgemäßen Gebrauchs kann in der Mangelhaftigkeit der Leasingssache oder darin liegen, daß diese dem Leasingnehmer nicht oder nur zum Teil zur Verfügung gestellt wird (Senatsurteil vom 1. Juli 1987 – VIII ZR 117/86 = WM 1987, 1131, 1133). Das Ausbleiben der nach dem Leasingvertrag geschuldeten sowie für den Betrieb der Gesamtanlage wesentlichen acht Hardwareteile (vier Terminals, drei Akustikkoppler und ein Typendrucker) stellt eine derartige nur teilweise Zurverfügungstellung dar.
3. Die einer Kündigung nach § 542 BGB grundsätzlich vorzuschickende befristete Aufforderung an den Schuldner zur vertragsgemäßen Gebrauchsverschaffung ist dann entbehrlich, wenn angesichts der verfloßenen Zeit und der damit einhergehenden technischen Veralterung und des Preisverfalles der Hardware an der weiteren Vertragserfüllung kein Interesse mehr besteht.