

Die Dokumentklasse juraabbrev*

Lucas Wartenburger

2004/09/15

Zusammenfassung

In (rechts-)wissenschaftlichen Dokumenten muss häufig ein Abkürzungsverzeichnis erstellt werden. Dabei ist darauf zu achten, dass alle verwendeten Abkürzungen, aber auch nur diese, in das Verzeichnis aufgenommen werden. Dies wird durch das vorliegende Paket erleichtert. Die Abkürzung kann an einer Stelle des Dokumentes (oder besser: zentral in einer wiederverwendbaren Datei) definiert und sodann überall verwendet werden. Dieses Paket unterstützt dabei in erster Linie solche Funktionen, die für deutsche juristische Texte hilfreich sein könnten.¹

Inhaltsverzeichnis

1	Einführung	2
2	Vorbereitung	2
2.1	Installation	2
2.2	Einbindung des Paketes	2
3	Erstellen eines Abkürzungsverzeichnisses	3
3.1	Definieren von Abkürzungen	3
3.2	Verwenden von Abkürzungen	4
3.3	Besonderheiten im Umgang mit Literatur	5
3.4	Abstand nach Abkürzungen	5
3.5	Zusammenfassung: Die „Anhängsel“ des <code>abb</code> -Befehls	8
3.6	Macros in Abkürzungen	9
4	Erstellung eines Gesetzesverzeichnisses	10
4.1	Dokument-Header	10
4.2	Gesetzeszitate	10
4.3	Gesetzesverzeichnis	11

*Version v0.51

¹Es kann aber durchaus auch in anderen Texten genutzt werden. Das statement oben habe ich an sich nur eingefügt, damit es keinen Grund gibt, die ganze Doku auf Englisch zu übersetzen.

1 Einführung

An sich ließe sich ein Abkürzungsverzeichnis auch mittels der Verwendung von `index.sty` oder ähnlicher Pakete erstellen. Ein solches Verzeichnis wäre sogar (dank `makeindex`) automatisch sortiert – eine Arbeit, die man im vorliegenden Paket selbst übernehmen muss. Der Aufruf der `index`-Kommandos ist jedoch relativ umständlich; ferner muss extra `makeindex` gestartet werden.

Das vorliegende Paket hat den Vorteil, dass jede definierte Abkürzung in drei Formen verwendet werden kann: als Abkürzung, als Langform oder in einem Befehl, der abhängig von der jeweiligen Umgebung entweder die Lang- oder die Kurzform erzeugt. Außerdem kümmert sich das Paket darum, dass der Zwischenraum nach der Abkürzung den Konventionen entspricht und erspart es dem Anwender dadurch, sich hin und wieder auf \TeX -Niveau zu begeben.

Wer schon die Vorversion benutzt hat, wird überrascht sein. Nicht nur der Name hat sich geändert, sondern auch sonst fast alles (v.a. hinter den Kulissen). Aber auch einige Befehle funktionieren nicht mehr. Dafür geht die Neufassung besser mit den Ressourcen um und ist wesentlich flexibler. Neu ist die Unterstützung von `hyperref` und \TeX 4ht.

2 Vorbereitung

2.1 Installation

Um die erforderlichen Dateien zu erzeugen, muss \TeX mit dem Aufruf

```
tex juraabbrev.ins
```

gestartet werden. Die Datei `juraabbrev.sty` muss so abgelegt werden, dass \LaTeX sie später findet. Die Dateien `abbrevtest.tex` sowie `laws.ist` dienen als Beispiel.

2.2 Einbindung des Paketes

Um `juraabbrev` zu verwenden, genügt eine Einbindung in den header des Dokuments

```
\usepackage{juraabbrev}
```

Wenn ein Gesetzesverzeichnis erstellt werden soll, muss die Datei `index.sty` vor `juraabbrev` eingebunden werden.

3 Erstellen eines Abkürzungsverzeichnisses

3.1 Definieren von Abkürzungen

`abbreviations` Abkürzungen werden dort definiert, wo später im Text das Abkürzungsverzeichnis erscheinen soll. Die Definitionen müssen sich in der Umgebung `abbreviations` befinden.² In der Umgebung `abbreviations` wird das Verzeichnis zweispaltig dargestellt. Der Titel des Abkürzungsverzeichnisses wird durch den Befehl `\abbrevname` definiert. Damit können – soweit das sinnvoll ist – auch mehrere Verzeichnisse erstellt werden, etwa so:

```
\begin{abbreviations}
  \abbrev{bgb}{BGB}{Bürgerliches Gesetzbuch}
  \abbrev{estg}{ESTG}{Einkommensteuergesetz}
  [...]
\end{abbreviations}
\renewcommand{\abbrevname}{Abgekürzt zitierte Literatur}
\begin{abbreviations}
  \abbrev{hhr}{HHR}{Herrmann/Heuer/Raupach}
  \abbrev{tl}{TL}{Tipke/Lang}
\end{abbreviations}
```

Das Ergebnis würde dann so aussehen (vorausgesetzt, alle Abkürzungen werden irgendwo in der Kurzform verwendet):

Abkürzungsverzeichnis

BGB	Bürgerliches Gesetzbuch
ESTG	Einkommensteuergesetz
z.B.	zum Beispiel
ca.	circa
a.a.O.	am angegebenen Ort

Abgekürzt zitierte Literatur

HHR	Herrmann/Heuer/Raupach
TL	Tipke/Lang

Für die weitere Verwendung der Abkürzung spielt es dann keine Rolle, in welchem Verzeichnis diese stehen.

`\abbrev` Die Definition einer Abkürzung erfolgt über den Befehl `\abbrev`. `\abbrev` hat folgenden Syntax:

²Wer die nicht gebrauchen kann, kann sie auch weglassen. Es müssen dann aber die Maßeinheiten `leftsize`, `middistance` und `rightsize` definiert werden.

`\abbrev{<Kennung>}{<Kurzform>}{<Langform>}`

z.B.: `\abbrev{dstr}{DStR}{Deutsches Steuerrecht}`

`\abbrev*` Die *-Version des `\abbrev`-Befehls bewirkt, dass der Eintrag in jedem Fall im Abkürzungsverzeichnis erscheint. So könnte man etwa Abkürzungen definieren, die in jedem Fall zig-mal im Dokument vorkommen (S., ff.), ohne dass man sie ständig mit einem speziellen Befehl aufrufen möchte. Ansonsten erscheint die Erläuterung im Verzeichnis nur, wenn die Abkürzung irgendwo im Dokument in der Kurzform verwendet wurde.

3.2 Verwenden von Abkürzungen

Auf die so definierten Abkürzungen kann man im Dokument (und zwar auch vor der Stelle, an der das Verzeichnis steht) recht leicht zurückgegriffen werden. Das Paket stellt dazu ein Kommando mit mehreren „Erweiterungssymbolen“ zur Verfügung. Der Befehl `\abb{<Kennung>}` erzeugt im Dokument die Kurzform der Abkürzung. `\abbgb` gibt also BGB zurück. Das Paket unterstützt auch `hyperref`. In einer PDF-Datei kann man dann per Mausklick von der Abkürzung zur Erläuterung im Abkürzungsverzeichnis gelangen.

Nur wenn der `\abb{<Kennung>}`-Befehl mindestens einmal im Dokument verwendet wird, erscheint die Abkürzung (nach dem übernächsten \LaTeX -Lauf) zwingend im Abkürzungsverzeichnis. Den Aufruf einer im vorherigen \LaTeX -Lauf noch nicht definierten Abkürzung quittiert \TeX mit einer Fehlermeldung, da der `\abb`-Befehl dann schlicht ein unbekanntes Kommando darstellt (zu Umgehungsmöglichkeiten siehe sogleich).

`\abb{<Kennung>+}` Der Befehl `\abb{<Kennung>+}` erzeugt entsprechend die Langform der Abkürzung. Er ersetzt den früheren Befehl `\labb{<Kennung>!}` `\abbgb+` erzeugt also: Bürgerliches Gesetzbuch. In diesem Fall wird die Abkürzung nicht ins Verzeichnis aufgenommen (sie wurde ja auch nicht verwendet). Man kann so also „inoffizielle“ Abkürzungen im Dokument verwenden, um sich bei komplizierten Begriffen nicht zu vertippen. (Um die Flexion muss man sich aber selbst kümmern! Vielleicht lernt `juraabbrev` das später auch noch.)

`\abb{<Kennung>*` Der Befehl `\abb{<Kennung>*` fungiert als Mischform zwischen den beiden anderen. Er ersetzt den früheren Befehl `\babb{<Kennung>!}` Der *-Befehl gibt normalerweise die Kurzform zurück, innerhalb einer `longabbreviations`-Umgebung jedoch die Langform. Zu Einsatzmöglichkeiten siehe sogleich.

Mal kurz: `\abbgb*`.

Mal lang: `\begin{longabbreviations}\abbgb*.\end{longabbreviations}`

erzeugt also:

Mal kurz: BGB.

Mal lang: Bürgerliches Gesetzbuch.

Ein Eintrag im Abkürzungsverzeichnis wird nur dann erstellt, wenn

`\abb⟨Kennung⟩*` zumindest einmal die Kurzform erzeugt hat.

`\abb⟨Kennung⟩<` Gelegentlich kommt es vor, dass eine Abkürzung mit einem kleinen Buchstaben beginnt (etwa „vgl.“). Wenn eine solche Abkürzung am Satzanfang steht, soll sie natürlich mit einem Großbuchstaben beginnen. Andererseits soll sie aber nicht doppelt im Verzeichnis erscheinen. Abhilfe schafft hier der Befehl `\abb⟨Kennung⟩<`. Die Eingabe `\abbca<` erzeugt also: Ca. (statt ca.).

`\abb` Die bisher genannten Befehle haben einen Nachteil: Wenn die Abkürzung erstmals verwendet wird, entsteht eine hässliche `TeX`-Fehlermeldung, da ein Makro aufgerufen wird, das `TeX` noch nicht kennt. Wer sich davor fürchtet, kann anstelle von `\abbbgb` auch `\abb_bgb_` benutzen. Die Leerzeichen sind in diesem Fall wirklich wichtig! Wer also etwas wie NJW-Fundstelle schreiben will, muss eingeben: `\abb_njw_Fundstelle`. Keine Angst, die Leerzeichen werden im Output ignoriert. Falls die Abkürzung noch gar nicht definiert wurde oder erstmals verwendet wird, bekommt man nun keine Fehlermeldung mehr, sondern nur eine Warnung

`\babb` `“Abbreviation ‘njw’ undefined”`. Die Befehle `\babb` und `\labb` funktionieren nach dem gleichen Schema, wobei `\babb` die *-Form erzeugt und `\labb` die +-Form (Langform).

3.3 Besonderheiten im Umgang mit Literatur

Für den `\babb`-Befehl gibt es an sich nur einen sinnvollen Einsatzbereich, nämlich die Literaturangaben. So kann man beispielsweise in eine `BIBTeX`-Datenbank eintragen:

```
... journal = {\abbnjw*}, ...
```

Das Literaturverzeichnis ließe sich dann in eine `longabbreviations` Umgebung einrahmen. Im Literaturverzeichnis würde dann erscheinen „Neue Juristische Wochenschrift“, in einer Fußnote dagegen „NJW“. Ein solches Vorgehen könnte sich v.a. bei ausländischen Zeitschriften empfehlen, damit der arme Leser nicht ein zwei verschiedenen Verzeichnissen nachschlagen muss, wo er den zitierten Aufsatz finden kann. Daneben wird es aber auch im Literaturverzeichnis Abkürzungen geben, die man nicht ausschreiben lassen möchte (etwa Gesetzesbezeichnungen im Titel eines Aufsatzes, oder Kürzel von Rechtsformen). Diese könnte man dann durch Verwendung von `\abb⟨Kennung⟩` (ohne *) vor der Langform schützen.

3.4 Abstand nach Abkürzungen

Dieser Abschnitt richtet sich nur an besonders Interessierte; man kann mit `juraabbrev` auch sehr gut ohne die Kenntnis der nachfolgend beschriebenen Details arbeiten.

Wer wirklich ein gutes Dokument erzeugen will, sollte auch auf die richtige Verteilung des „glue“ im Text achten.³ Spezielle Probleme stellen sich hier dann, wenn die Abkürzung mit einem Punkt endet (etwa „z.B.“). Normalerweise würde `TeX`

³Sofern man mit `frenchspacing` arbeitet, ist dieser Aspekt ohnehin zu vernachlässigen.

diesen Fall so behandeln: Endet die Abkürzung mit einem Punkt nach einem Großbuchstaben (wie hier im Beispiel), wird dies nicht als Satzende angesehen und nur der normale „interword-glue“ wird eingefügt. Endet die Abkürzung dagegen mit einem Kleinbuchstaben („etc.“), so geht T_EX davon aus, dass hier der Satz zu Ende ist. Der Abstand wird bei Bedarf weiter gedehnt (normalerweise mit dem Faktor 3). Die Definition in `juraabbrev` sorgt dafür, dass der „space factor“ am Ende der Abkürzung auf 1000 steht. Das hat folgende Konsequenzen:

- Hat die Abkürzung selbst *keinen* Punkt und endet danach der Satz („So steht es im BGB.“), dann wird der spacefactor durch den Punkt *nach* der Abkürzung entsprechend den T_EX-Vorgaben hochgesetzt. Das im T_EXbook (Ch. 12) beschriebene Problem mit einem Satzende nach Großbuchstaben besteht nicht (man braucht also nicht zu schreiben `\abbbgb\null.`, sondern nur `\abbbgb`.⁴)
- Endet die Abkürzung mit einem Punkt, dann geht stellt `juraabbrev` den „space factor“ so ein, dass T_EX hier nicht von einem Satzende ausgeht.
- `\abb<Kennung>` • Schwierig wird es nur dann, wenn die Abkürzung mit einem Punkt endet und dort tatsächlich auch der Satz zu Ende sein soll. (Bei „etc.“ oder „usw.“ könnte das mal vorkommen.) Hier versteht T_EX nicht, dass ein Satz zu Ende ist, denn man wird ja nach der Abkürzung nicht noch einen Punkt setzen. Abhilfe schafft der Befehl `\abb<Kennung>` also z.B. `\abbao>`

Das war jetzt viel Theorie. Die Effekte sollen anhand der folgenden Beispielen kurz dargestellt werden. In den nachfolgenden Beispielen wurde jeweils die Zeile „a)“ normal eingegeben, bei Zeile „b)“ wurde `juraabbrev` verwendet. Die Texte werden anschließend reichlich gedehnt, damit der Unterschied sichtbar wird.

1. Beispiel: Nur mit `juraabbrev` erkennt T_EX das Satzende nach einem Grossbuchstaben.

```
Text a: „Das□steht□im□BGB.□Oder□etwa□nicht?“
Text b: „Das□steht□im□\abbbgb.□Oder□etwa□nicht?“
```

- Normal:
 - a) Das steht im BGB. Oder etwa nicht?
 - b) Das steht im BGB. Oder etwa nicht?
- 10 Punkte gedehnt:
 - a) Das steht im BGB. Oder etwa nicht?
 - b) Das steht im BGB. Oder etwa nicht?
- 20 Punkte gedehnt:
 - a) Das steht im BGB. Oder etwa nicht?
 - b) Das steht im BGB. Oder etwa nicht?

⁴Abgesehen davon würde das mit dem `null` auch gar nicht funktionieren, es würde dann ein Leerzeichen vor dem Punkt entstehen.

- 30 Punkte gedehnt:
 - a) Das steht im BGB. Oder etwa nicht?
 - b) Das steht im BGB. Oder etwa nicht?
2. Beispiel: Mit `juraabbrev` erkennt \TeX , dass mit der Abkürzung kein Satz endet.

Text a: „Der_Baum_ist_ca._10_Meter_hoch.“

Text b: „Der_Baum_ist_\abbca_10_Meter_hoch.“

- Normal:
 - a) Der Baum ist ca. 10 Meter hoch.
 - b) Der Baum ist ca. 10 Meter hoch.
 - 10 Punkte gedehnt:
 - a) Der Baum ist ca. 10 Meter hoch.
 - b) Der Baum ist ca. 10 Meter hoch.
 - 20 Punkte gedehnt:
 - a) Der Baum ist ca. 10 Meter hoch.
 - b) Der Baum ist ca. 10 Meter hoch.
 - 30 Punkte gedehnt:
 - a) Der Baum ist ca. 10 Meter hoch.
 - b) Der Baum ist ca. 10 Meter hoch.
3. Beispiel: Hier ändert sich nichts, da \TeX nach einem Grossbuchstaben ohnehin kein Satzende annimmt.

Text a: „Das_sagt_z.B._einer_meiner_verehrten_Kollegen.“

Text b: „Das_sagt_\abbzb_einer_meiner_verehrten_...“

- Normal:
 - a) Das sagt z.B. einer meiner verehrten Kollegen.
 - b) Das sagt z.B. einer meiner verehrten Kollegen.
- 10 Punkte gedehnt:
 - a) Das sagt z.B. einer meiner verehrten Kollegen.
 - b) Das sagt z.B. einer meiner verehrten Kollegen.
- 20 Punkte gedehnt:
 - a) Das sagt z.B. einer meiner verehrten Kollegen.
 - b) Das sagt z.B. einer meiner verehrten Kollegen.
- 30 Punkte gedehnt:
 - a) Das sagt z.B. einer meiner verehrten Kollegen.
 - b) Das sagt z.B. einer meiner verehrten Kollegen.

4. Beispiel: Wenn ein Satz mit einer Abkürzung endet, die einen Punkt enthält, muss der Befehl `\abb⟨Kennung⟩>` verwendet werden, um das Problem zu umgehen.

Text a: „So□schon□Müller,□a.a.O.□Neuerdings□...“

Text b: „So□schon□Müller,□\abb \langle aa \rangle □Neuerdings□...“

- Normal:
 - a) So schon Müller, a.a.O. Neuerdings anders aber Meier.
 - b) So schon Müller, a.a.O. Neuerdings anders aber Meier.
- 10 Punkte gedehnt:
 - a) So schon Müller, a.a.O. Neuerdings anders aber Meier.
 - b) So schon Müller, a.a.O. Neuerdings anders aber Meier.
- 20 Punkte gedehnt:
 - a) So schon Müller, a.a.O. Neuerdings anders aber Meier.
 - b) So schon Müller, a.a.O. Neuerdings anders aber Meier.
- 30 Punkte gedehnt:
 - a) So schon Müller, a.a.O. Neuerdings anders aber Meier.
 - b) So schon Müller, a.a.O. Neuerdings anders aber Meier.

3.5 Zusammenfassung: Die „Anhängsel“ des `abb`-Befehls

Auf den ersten Blick sehen die verschiedenen Möglichkeiten, den Befehl `\abb⟨Kennung⟩` aufzurufen, sicher etwas verwirrend aus. Die „Sonderzeichen“ wurden aber mit Bedacht gewählt.

- + erzeugt die Langform (man bekommt „mehr“, deshalb +)
- < erzeugt den Großbuchstaben am Satzanfang (öffnendes Zeichen, da der Satz beginnt)
- > erzeugt den Zusatzabstand am Satzende (schließendes Zeichen, da Satz endet)
- * erzeugt die wandelbare Abkürzung (weil es was „Besonderes“ ist, mit Sternchen)

Man kann diese Zeichen beliebig miteinander kombinieren, verketteten und wiederholen. Wichtig ist nur, dass sich die vier erlaubten Sonderzeichen direkt an den Befehl anschließen. Kommt ein anderes Zeichen dazwischen, haben die nachfolgenden Symbole keinen Einfluss mehr auf die Abkürzung. Der Befehl `\abbvg1*<>` würde also die Abkürzung „Vergleiche“ oder „Vgl.“ ausgeben (je nach Umgebung, aber immer mit Großbuchstaben beginnend) und ein nachfolgendes Leerzeichen wie nach einem Satzende dehnen. Eine Kombination von „+“ und „*“ ist möglich, bringt aber nichts, da aufgrund des „+“ immer die Langform ausgegeben wird.⁵

⁵ Für Bastler: Die vier „Sonderzeichen“ lassen sich auch durch einen einfachen Eingriff in das package verändern. Wichtig ist nur, dass diese Zeichen zur Kategorie 12 gehören – also keine Buchstaben! Die Zeichen wurden aber absichtlich so gewählt, dass nicht davon auszugehen ist, dass eines dieser Zeichen unmittelbar nach einer Abkürzung gebraucht wird, anders als z.B. bei einem Bindestrich (BMF-Schreiben!).

3.6 Macros in Abkürzungen

`\protectab` Eine letzte Besonderheit gibt es noch zu erklären. Es kann vorkommen, dass man eine Abkürzung im Langtext einer Abkürzung verwenden möchte. (etwa: GmbHR entspricht GmbH-Rundschau). In der Liste sollte dann natürlich die Erklärung für GmbH auch erscheinen. Man kann die `\abb`-Befehle auch im 3. Argument des `\abbrev`-Befehls verwenden, aber man muss (wenn man nicht mit Fehlermeldungen bombardiert werden will) den Befehl `\protectab` voranstellen. Wenn nach der „inneren“ Abkürzung ein Leerzeichen kommt, muss man auch dies extra kennzeichnen durch `_`. Ein Beispiel:

```
\abbrev{gmbh}{GmbH}{Gesellschaft\_mit\_beschränkter\_Haftung}  
\abbrev{gmbhr}{GmbHR}{\protectab\abbgmbh-Rundschau}
```

ergibt „GmbH-Rundschau“ oder

```
\abbrev{gmbhr}{GmbHR}{\protectab\abbgmbh\_Rundschau}
```

ergibt „GmbH Rundschau“.

Dieses Feature lässt sich auch nutzen, wenn man andere Kontrollsequenzen oder active characters in einer Abkürzung benutzen möchte. Denkbar wäre folgende Deklaration:

```
\abbrev{llm}{\protectab\scshape{}ll.m.}{Master of Laws}  
\abbrev{drjur}{Dr.\protectab~iur.}{Doctor iuris}  
\abbrev{ff}{\protectab\s1{}ff.}{folgende}  
[...]
```

Das Ergebnis:

Abkürzungen

LL.M.	Master of Laws
Dr. iur.	Doctor iuris
<i>ff.</i>	folgende
GmbH	Gesellschaft mit beschränkter Haftung
GmbHR	GmbH-Rundschau
S.	Seite

Daraufhin kann man dann solch schöne Sätze kreieren wie:

```
Ein wichtiger Aufsatz von \abbrjur Heins Hansen, \abblm (NYU),  
\abbgmbhr 2000, \abbs~15 \abbff>
```

Also: „Ein wichtiger Aufsatz von Dr. iur. Heins Hansen, LL.M. (NYU), GmbHR 2000, S. 15 *ff.*“⁶

⁶Die NYU wurde hier nicht als Abkürzung definiert. Man könnte das natürlich auch tun und dort ggf. auch gleich die Kapitälchen einstellen. Aus guten Gründen beschränkt sich die Wirkung

4 Erstellung eines Gesetzesverzeichnisses

Die `\abb`-Funktionen lassen sich selbstverständlich auch für die gängigen Abkürzungen von Gesetzen verwenden. Wer darüber hinaus ein Verzeichnis der zitierten Gesetze haben möchte, kann dies mit dem vorliegenden Paket und dem `index`-Paket recht leicht bewerkstelligen.

4.1 Dokument-Header

Zunächst müssen die Pakete eingebunden werden.

```
\usepackage{index}
\usepackage{juraabbrev}
```

Anschließend muss ein Index nach folgendem Muster erstellt werden (noch vor `\begin{document}`!):

```
\newindex{laws}{ldx}{lnd}{Zitierte Gesetze}
```

Wichtig ist dabei v.a. die Kennung `laws`, der Rest kann abgewandelt werden, wenn man den Aufruf von `makeindex` entsprechend ändert.

4.2 Gesetzeszitate

`\citelaw` Der wichtigste Befehl zum Zitat eines Gesetzes lautet `\citelaw`. Der Syntax ist:
`\citelaw[Paragraphenzeichen][Paragraph][Absatz][Satz][Gesetz]`

Der Parameter `Gesetz` enthält dabei die Kommandokennung aus dem Abkürzungsverzeichnis, also z.B. „bgb“. Der Aufruf

```
\citelaw[\S]{15}[Abs. 1][S. 1 Nr. 2]{estg}
```

erzeugt daher folgende Ausgabe:

§ 15 Abs. 1 S. 1 Nr. 2 EStG.

Zugleich wird ein Indexeintrag der folgenden Form erstellt:

```
\index[laws]{estg@\abbestg*!\S~15!Abs.~1}
```

Zur Bedeutung dieses Eintrages sei auf die Doku von `index.sty` verwiesen. Die *-Form des Befehls erzeugt keine Ausgabe, sondern nur einen Verzeichniseintrag.

`\citepar` `\citeart` Da das „Paragraphenzeichen“ in der Regel „§“ oder „Art.“ lautet, gibt es zwei
der Kapitälchen auf die Abkürzung selbst.

Wer genau hingeschaut hat, hat auch gemerkt, dass die Tilde nicht in die Definition von `S` eingebunden wurde. Das Problem dabei ist, dass automatisch nach jeder Abkürzung ein Leerzeichen erzeugt wird. `juraabbrev` achtet dabei (noch) nicht darauf, ob die Abkürzung selbst evtl. schon mit einem Leerzeichen endet. Daher funktioniert diese Vereinfachung – zumindest bislang – nicht.

Die Definition von so vielen Abkürzungen mag überflüssig erscheinen. Der Text gewinnt aber gerade durch die Spacing-Überwachung mit `juraabbrev` an Qualität. Die Frage, ob eine Häufung von so vielen Abkürzungen in einem Satz sinnvoll ist, soll hier aber nicht vertieft werden. Wer `TEX` zum Wahnsinn bringen möchte, kann ja mal seinen Palandt mit `juraabbrev` abschreiben!

Kurzformen des Befehls: `\citepar` und `\citeart`.

```
\citepar{<Paragraph>}[<Absatz>][<Satz>]{<Gesetz>}
\citeart{<Artikel>}[<Absatz>][<Satz>]{<Gesetz>}
```

Auch diese verfügen jeweils über die *-Form. Den `\citelaw`-Befehl wird man daher nur noch in Ausnahmefällen, etwa für ausländische Gesetze, benötigen.

4.3 Gesetzesverzeichnis

Um das Gesetzesverzeichnis zu erstellen, muss `makeindex` wie folgt aufgerufen werden:

```
makeindex -o <Dateiname>.lnd <Dateiname>.ldx -s laws.ist
```

Das Verzeichnis erscheint im Dokument an der Stelle des `\printindex`-Befehls. Dieser kann in ein `longabbreviations`-Environment eingefügt werden, damit die Gesetze im Verzeichnis ausgeschrieben werden.

```
\begin{longabbreviations}
\printindex[laws]
\end{longabbreviations}
```

5 Kontakt

Das Paket ist sicher noch nicht ausgereift und wird in der nächsten Zeit noch erweitert und verbessert werden. Wer Vorschläge dazu hat, kann sich gerne an mich wenden.

Folgende Probleme sind mir schon aufgefallen:

- wer `hyperref` verwendet, kann keine `\abb`-Befehle in Überschriften einsetzen. In der `hyperref`-Doku ist beschrieben, warum das nicht geht und wie man die Probleme umgehen kann. Ich bin noch nicht dazu gekommen.
- doppelte Leerzeichen, wenn Abkürzung mit `space` endet (oben beschrieben)
- aus der Interaktion von `hyperref` und `index.sty` ergeben sich Fehler, wenn das Dokument nicht mit `PDF \TeX` , sondern normal mit `L \TeX` bearbeitet wird.
- keine Links bei exportierten Dateien (`TeX4ht`)

Ich bin zu erreichen unter:

Lucas Wartenburger <elwwa@gmx.net>