

# LEX<sub>TEX</sub>: Context-Sensitive Legal Citations for L<sub>TEX</sub>

Frank G. Bennett, Jr.\*

Law Department  
School of Oriental and African Studies  
Thornhaugh Street  
London WC1H 0XG  
Internet: fbennett@c1us1.u1cc.ac.uk

## Abstract

The most widely used style guide for legal publications in the United States is *A Uniform System of Citation* (the so-called 'Blue Book'). LEX<sub>TEX</sub> is a L<sub>TEX</sub> style which automates typeface selection, citation truncation, citation cross-referencing, and the production of tables of authorities in the Blue Book style. This article discusses the design problems encountered in coding LEX<sub>TEX</sub>, points out a few of the dirty tricks which were used to smarten up its response to context, and comments on further work that waits to be done in this area. A reference card for use with LEX<sub>TEX</sub> is included.

## The Blue Book

Law reviews in the United States are a curious corner of the publishing world. Traditionally, most such journals are edited entirely by law students, who compete in their first year of study for admission to staff positions. Participation in law review, particularly in an editorial post, is viewed by potential employers as an important token of accomplishment in making hiring decisions. And so it goes that despite the long hours and lack of pay, the competition for these positions is quite intense.

The most widely used style in U.S. law reviews is *A Uniform System of Citation*<sup>1</sup> (the so-called 'Blue Book'),<sup>2</sup> compiled and periodically revised by staff at the Columbia Law Review, the Harvard Law Review Association, the University of Pennsylvania Law Review and the Yale Law Journal. The Blue Book requires that the origin of each proposition asserted or referred to by the author be precisely identified in

\* The author wishes to thank Boris Pevzner and Steve Bellovin for their patience with and comments on the initial beta 1.0 release of LEX<sub>TEX</sub> and its friends. The (as yet unrealized) BIB<sub>TEX</sub> analogue discussed under Further Work was suggested by Steve Bellovin. Gratitude is also due to George Greenwade for his encouraging words on the initial release, and to Andrew Hoddinott for introducing me to T<sub>EX</sub> in the first place.

<sup>1</sup> A UNIFORM SYSTEM OF CITATION (14th ed. 1991).

<sup>2</sup> With the emphasis on 'blue'. With the opposite stress, the term refers to a rather different type of literature.

a footnote citation. It lays down detailed and specific rules concerning typeface conventions, short-form citations and cross-referencing. All of these rules vary according to the nature of the source as well as the context in which the citation appears, which makes the proofreading of legal citations a particularly unenviable chore.

Indeed, in the eyes of some, the Blue Book's rules are specifically tailored to test the persistence and will power of those lucky enough to find their waking lives revolving around its creed. Nonetheless, there is enough underlying consistency in its conventions that typeface selection and the formatting of short-form citations can be almost completely automated. The key to context sensitivity as required by the Blue Book is the ability to remember key portions of the text on a start-to-finish reading. Both law review editors and T<sub>EX</sub> are capable of doing this; the difference is that T<sub>EX</sub> is a good deal faster, and rather less likely to develop indigestion.

The Reference Card at the end of this article, together with the tables illustrating the use of style blueprints and citation templates, should contain all of the information necessary for ordinary use of LEX<sub>TEX</sub>. The flow diagram in the Reference Card should help the user to visualize how the style operates. If I have accomplished my original purpose in writing the code of LEX<sub>TEX</sub>, it should be possible for a reader who is generally familiar with both L<sub>TEX</sub> and the Blue Book to make a copy of the Reference Card and the tables, read no further than the end of this sentence, and immediately begin using LEX<sub>TEX</sub> productively in his or her own work. This should be

the first and final test of L<sup>A</sup>T<sub>E</sub>X's utility. The Blue Book is bad enough by itself; if make-easy software for it requires a lengthy manual, one might as well not bother.

On the slightly reckless assumption that I *have* accomplished my original purpose, the scope of this article is limited to a general overview of the design of L<sup>A</sup>T<sub>E</sub>X, with a few ancient mariner's tales of problems that cropped up in the course of simultaneously drafting the style and learning the rudiments of the T<sub>E</sub>X language. The following section explains how L<sup>A</sup>T<sub>E</sub>X attempts to reduce the citation formatting process to a manageable set of units. This is followed by a discussion of some of the more entertaining problems dealt with by the code. The article closes with some comments on further work that is waiting to be done in this line.

### The L<sup>A</sup>T<sub>E</sub>X Style Engine

The format of any Blue Book citation can be fully described in terms of six essential *text units*, and nine punctuation *bridges* (seven, actually, two of which have a plural 'alter ego', for a total of nine in all). The text units each require up to two associated typefaces, which I will refer to as the *main* and *alternative* typefaces for each text unit. The presentation form of any Blue Book citation can be described in terms of text units, associated typefaces, and bridges; and the various mutations of the citation in subsequent references can be accomplished by manipulating units and bridges, without reference to their actual contents. This is what makes the automation of citations possible.

In order to combine flexibility with ease of use in an automated system,<sup>3</sup> we can (1) define a macro which, from a blueprint of necessary details, (2) defines another macro which in turn, from a set of text arguments, (3) defines macros to stand for individual citations. In step (2), only the text need be fed to the macro; everything else can have been

<sup>3</sup> It should be mentioned that L<sup>A</sup>T<sub>E</sub>X is not the first attempt to automate short-form citations. See the materials prepared by David Rhead in connection with the L<sup>A</sup>T<sub>E</sub>X3 project: Rhead, *Towards B<sub>B</sub>T<sub>E</sub>X style-files that implement principal standards*, T<sub>E</sub>X LINE 12 (May 1990); Rhead, *How might L<sup>A</sup>T<sub>E</sub>X3 deal with citations and reference-lists*, T<sub>E</sub>XLINE 13 (September 1991). The code behind this proposed interface, as I understand it, defines each style of citation separately. Through the use of blueprints and the style template macro `\@law@newcitestyle`, L<sup>A</sup>T<sub>E</sub>X attempts to provide a more general, customizable interface.

#### Six essential text units

1. The author's name;
2. The title of the work;
3. The source in which the cited work is located, including any volume number;
4. The page on which the cited work is located in the source;
5. An optional reference to a particular page or section number of the work; and
6. The tail end of the citation (usually a year of publication, perhaps with an indication of the editor, translator, publisher and city of publication as well, all usually enclosed in parentheses).

#### Nine punctuation bridges

1. Author-to-Title
2. Title-to-Work
3. Work-to-Source
4. Source-to-Location-page (singular and plural)
5. Location-page-to-Specific-reference-page
6. before-the-Tail-end
7. after-*Id./supra* (singular and plural)

defined in advance. After once feeding the details of a particular source to a macro in this way, it should be possible to reduce subsequent references to it to a single, context-sensitive nickname.

To state this in shorthand terms, the first, most general macro in this series can be thought of as a 'style template', and the macros defined by it as 'citation templates'. The citation templates produce 'nickname macros', which in turn are used to produce actual printed citations in the form appropriate to a given context.

The blueprints for use by the style template are all contained in the file `lexicite.tex`. Additional blueprints can be added to this file by the adventurous user as required. In normal circumstances, the user need only determine which citation template is appropriate to a given citation, and feed it the text of the citation as a set of arguments, along with a nickname that will stand for the citation in subsequent references.

### Fun in the Sun

Once this basic structure is in place, it is largely a matter of fine-tuning to bring the output of the package into line with Blue Book requirements. This section discusses some of the tricks and kludges used to that end. It is not a comprehensive discussion,

Sample `lexcite.tex` blueprint entry for L<sup>A</sup>T<sub>E</sub>X

```

\newcitestyle{newarticlea}%
{Typefaces:author(x)b:title(i)b:cite(s)b/Citetype(a)c}%
{Bridges:author(\ )dtitle(\ )dcite(\ / \ )dpage(\ )drefpage(\
)dafterid(\ at~\ at~)d}%
{6}e%
{{#1}f{#2}f{#3}f{#4}f{#5}f{(#6)}f}%

```

<sup>a</sup> This gives the name of the citation template to be generated.

<sup>b</sup> Within these sets of parentheses, for the author, title and first-cite-part portions of the citation, *s*, *b* and *i* select small caps, boldface and italic type as the main typeface. *S*, *B* and *I* respectively select each of these as a typeface which may be specially selected using `\...\`. The parentheses may *not* be left empty. To use the default (roman) typeface, place some other character here.

<sup>c</sup> Within this set of parentheses, *a*, *b*, *c* and *s* will classify all citations generated by this citation template as being to articles, books, cases or statutes, respectively. These parentheses must *not* be left empty.

<sup>d</sup> These parentheses contain the citation bridges that will be placed between the portions of the citation indicated. Two bridges, singular and plural, are separated by a / in the parentheses following `cite` and `afterid`. The bridges following `afterid` are used to attach specific page references to *Id.* and *supra* citations. All of these parentheses may be left empty (provided that the plural separator / is not omitted from the two fields to which it applies).

<sup>e</sup> This argument states how many arguments the finished citation template will accept. It must be a number between 1 and 6.

<sup>f</sup> Six pairs of matched braces must appear inside this argument. Any other than the first (which represents the citation nickname) may be left empty, but the bridges must of course take this into account. The number of arguments inserted here must correspond with the number stated on the line above.

but may help those interested in making improvements or alterations. Those wishing to dig deeper on the technical side are invited — indeed, positively encouraged — to have a look at the code itself, which is available from the major T<sub>E</sub>X archives. Those who want a friendlier introduction to the care and feeding of L<sup>A</sup>T<sub>E</sub>X are gently referred to the penultimate paragraph of the first section of this article — after which they are invited to suggest improvements.

**Eliminating redundancy.** When I began drafting the code of this style, the file `lexitex.sty` quickly grew to about 30 kilobytes, roughly the same size as in the current release. But in its early life, L<sup>A</sup>T<sub>E</sub>X had separate storing and printing routines for each of the four general classes of citation; and as new features were added it became increasingly cumbersome to carry the changes through to each set of routines.

Eventually the pressure of common sense overwhelmed me, and these separate routines were collapsed into a single set of macros which can respond differently in minor ways depending upon a toggle indicating the citation type. While this is little more than good programming practice, it was surprising to see the extent to which a job that initially seemed

to bristle with nasty real-world edges could be distilled into a compact and unified logical structure.

The code will no doubt tolerate a good deal more optimization and trimming.

**Expansion control.** While it is processing a job, L<sup>A</sup>T<sub>E</sub>X swaps a great deal of information between macros. Much of this ends up being printed in a variety of L<sup>A</sup>T<sub>E</sub>X environments, or is exported to external files. Rigorous expansion control is therefore crucial if undesired smash-ups are to be prevented. And in some places `\noexpand` and `\the` statements were not enough by themselves to prevent gastric chaos.

To cope with such situations, L<sup>A</sup>T<sub>E</sub>X provides the `\@law@clean` macro, which accepts a token register and an arbitrary control sequence as its arguments. When used, it first defines a number of common control sequences which expand into T<sub>E</sub>X primitives (which cause difficulties if expanded at certain crucial stages of L<sup>A</sup>T<sub>E</sub>X's ruminations) as strings of themselves: `\string\LexiTeX`. The definition is stored under the arbitrary control sequence name *locally*, so that its scope can be limited by grouping. The control sequence can then safely be used to stand for the token register in L<sup>A</sup>T<sub>E</sub>X's internal code. The `\@law@clean` macro runs `\@law@clean over`

all token registers which might contain problematic control sequences. If special character or formatting commands within a L<sup>A</sup>T<sub>E</sub>X argument give an error during processing, this may be due to the absence of a T<sub>E</sub>X primitive invoked by this command from the list of temporary definitions in `\@law@clean`. If adding a suitable entry covering the command to the list makes the problem go away, please send me the details, and I will include the patch in future releases.

**Output transformations.** The citation templates produce two macros for each citation, `\<blah>` and `\<blah>full`, for short- and full-form citations respectively. The `\<blah>full` macro is invoked automatically immediately after a citation is declared using the citation template, unless output is suppressed by the user. Afterwards, the short-form `\<blah>` will normally be used; the `\<blah>full` form can be quietly ignored, but it is there for what it is worth.

Short-form output in the text must be context-sensitive. For most governing conditions, this is pretty pedestrian: a variable or a macro marks the occurrence of an event, such as a footnote number or the name of the immediately preceding citation, and the short-form print routine is set up to adjust the elements of the citation accordingly. One Blue Book rule does present special difficulties, however.

In short-form references to articles and books for which an author is specified, the title or name of the work should be omitted (automatically, that is), *unless some other work by the same author in the same typeface has been cited already*. The problem is that L<sup>A</sup>T<sub>E</sub>X's citation macros can only be expanded one at a time, and they are designed to do just one thing—make a citation *now*. An actual search for a matching author would be slow, and it would require significant and cumbersome re-design of portions of the code. Not fun.

In the end there was a much simpler solution. The author's name *itself* can be defined as a macro, by first running the register containing the author's name through `\@law@clean`, and then using `\csname\endcsname` to make a macro of it. This routine is made part of the citation template. The first time around, it is defined to expand to 1, but if it is found to be defined already, it is set to expand to 2. With this information, the print routine can honor the Blue Book rule without knowing which nickname citation is associated with the matching author name.

**Looking ahead for signals.** The one element of a citation that cannot be stored by the citation tem-

plate is the reference to the specific page or section number where a particular proposition occurs in the cited work. This page number must be added after the nickname macro is created, and it must be incorporated into the citation before it is printed. The goal is to adopt a syntax where:

`\bluebook+{21}`.

will expand into the full citation form:

A UNIFORM SYSTEM OF CITATION 21 (1991).

or into the appropriate short form:

A UNIFORM SYSTEM OF CITATION, *supra* note 1 at 21.

This requires that the nickname macro look ahead in the document before it prints its contents. For the most part this is straightforward. After resetting the appropriate registers to reflect the fixed information contained in the citation, nickname macros look ahead with L<sup>A</sup>T<sub>E</sub>X's `\@ifnextchar` command to see if the character coming up satisfies any of a series of conditions. If the next character is `*` or `-`, the print routine is bypassed, and printing is suppressed.<sup>4</sup> If the character is `+`, L<sup>A</sup>T<sub>E</sub>X gobbles the `+`, and checks further to see if there is a second `+`. If there is, it knows that the upcoming argument refers to multiple pages or sections. If there is not, L<sup>A</sup>T<sub>E</sub>X checks for a `-`, which would indicate multiple references with the use of a singular bridge (as in Federal Constitution sch. 5, paras. 2 & 4 (1957)). If *that* is not present, then the reference must be to a single page or section.

From L<sup>A</sup>T<sub>E</sub>X's own command syntax, it was clear that this kind of conditional testing is possible in T<sub>E</sub>X, but before I could implement it in L<sup>A</sup>T<sub>E</sub>X I needed to learn a fairly simple lesson about how T<sub>E</sub>X churns through a document. The difficulty I experienced is one that people who, as was my own case, come to T<sub>E</sub>X from higher-level procedural languages are likely to experience, so perhaps it is worth mentioning.

The L<sup>A</sup>T<sub>E</sub>X command `\@gobble` is a simple macro that accepts one argument and does nothing with it: `\def\@gobble#1{}`. It is necessary to ensure that `\@gobble`, when used, expands *immediately* before the character or argument that is to be dispensed with. But the T<sub>E</sub>X language does not have a means of distinguishing between the document 'itself' and the macros that expand within it; it's all so much program code. A complex set of macros such as L<sup>A</sup>T<sub>E</sub>X therefore requires careful structuring to ensure that

<sup>4</sup> This feature might be required where a non-conforming citation is used for a cited work, such as citations which are part of a quotation.

a `\@gobble` statement is the very *last* operation performed in any chain of macros leading up to it. In particular, it cannot be nested within an `\if... \fi` condition, because it would chew up part of the condition itself, causing perhaps all sorts of strange and wonderful things to happen.

L<sup>A</sup>T<sub>E</sub>X's `\@ifnextchar` hack works by storing either the first or the second of the two alternative arguments shown to it in a particular macro, and expanding this stored internal macro as its last act. L<sup>A</sup>T<sub>E</sub>X's `\@law@argscheck` routine uses `\expandafter` within the appropriate alternative argument to `\@ifnextchar` to run `\@gobble` as its 'last' act, before invoking the next command that needs to be chained. And so on back to the level of the document itself.<sup>5</sup>

Because the text of the initial full citation produced by the citation templates is actually written by the `\<blah>full` macro that they themselves define as their own last act, `+{<page number>}` can be added at the end of a citation template declaration as well as after a nickname reference.

This overall scheme reduces the typing required to specify page numbers to a minimum, and allows the same syntax to be used for this purpose for all types of citations. That in turn helps increase the speed with which a document can be prepared.

**Producing tables.** Once the details of a citation have been stored in a macro, it is a (relatively) simple matter to send it to an external table for processing by `makeindx`.

Every citation in L<sup>A</sup>T<sub>E</sub>X is flagged as being to a book, an article, a case or a statute. You can use L<sup>A</sup>T<sub>E</sub>X to produce lists of all citations falling into any of these four classes, simply by placing any of `\makebooktable`, `\makearticletable`, `\makecasetable`, or `\makestatutetable` before `\begin{document}`. This produces the files `*.btb`, `*.atb`, `*.ctb` and `*.stb` respectively.

Each of these files must then be sorted, using the `makeindx` utility. The `makeindx` program should be instructed to use the appropriate in-

<sup>5</sup> Those following the trail back upstream in the code will find that it goes cold at this point. This is because `\@law@argscheck` is inserted into the nickname macro by the citation template by first storing it to a temporary token register `\@tok@a` and using `\the` to invoke it. This is necessary because it falls within the scope of an `\xdef` that is necessary in order to "freeze" the current registers and macros in the nickname's definition. The use of `\the` prevents `\@law@argscheck` from actually expanding during this definition.

dex style file for that type of table. These are, respectively, `lexibook.ist`, `lexiarti.ist`, `lexicase.ist` and `lexistat.ist`. The output of each sort should be directed, respectively, to `*.bok`, `*.art`, `*.cas` or `*.sta`.<sup>6</sup> Finally, the command or commands `\printbooktable`, `\printarticletable`, `\printcasetable` or `\printstatutetable` should be placed where the tables are to appear when L<sup>A</sup>T<sub>E</sub>X is run over the document a final time.

Apart from producing tables for actual publication, the automatic generation of tables allows citations of each class to be gathered into a single location for proofreading. This can save a great deal of time and eyestrain.

**More expansion control.** For reasons which I readily admit to understanding only hazily, L<sup>A</sup>T<sub>E</sub>X stubbornly refused to cooperate when I attempted to place macros or `\the` statements inside a file-write macro based on L<sup>A</sup>T<sub>E</sub>X's own index-writing routine. The difficulty was related to expansion; certain control sequences apparently change their meaning between the time the file-write macro is run and the time of the actual export via L<sup>A</sup>T<sub>E</sub>X's output routine.

It seemed as though I had a choice of either suppressing expansion altogether (in which case the information in the registers or macros was often replaced before the write statement actually went into operation at the end of the page—resulting in a string of citations to the last source on the text page), or using `\immediate` to avoid the strange collision with the output routine (in which case the current page number written to the table might not be correct).

The solution adopted in the end was to use the `\@law@cleanup` macro, discussed above under the subheading Expansion Control. `\@law@cleanup` is a bit of a kludge, but it has the advantage of allowing us to selectively control the expansion of macros unknown to L<sup>A</sup>T<sub>E</sub>X without letting them boil themselves down to incomprehensible primitives before they end up in the external file.

**Sorting the citations.** Some materials need to be sorted in special ways, which cannot be derived easily from the citation. For example, some decisions of foreign courts ought to be sorted first by the name of the court, then by the date of the decision. Edited books, too, should be sorted by the name of the editor—which, in the Blue Book style, occurs in the tail end portion of the citation.

<sup>6</sup> This has only been successfully tested with the DOS version of `makeindx` that is supplied with `emTEX`. The long lines generated by L<sup>A</sup>T<sub>E</sub>X may cause problems for some versions of `makeindx`.

The easy solution adopted for this problem is to include the nickname of each citation in the output to the table file. The nickname is unique, and the citations therefore come out in the desired order, if the nicknames were chosen with this sorting process in mind. The macro of the form `\lexicite{<blah>}` can be used to invoke a citation nickname that contains a number, such as a date. When the sorted file is read by the relevant `\print*table` command, the nickname is ignored.

**Selective appearances.** The user may wish to selectively control the appearance of certain information in the citations and tables. For this purpose, L<sup>A</sup>T<sub>E</sub>X uses `^` and `_` as active characters.<sup>7</sup> The syntax for these is given in the reference card, and the principle of their use is straightforward. In all of the five main citation elements other than the nickname, the form `^...^` may be used to enclose text that will appear in the full form of the citation, and be eliminated both in the tables and in subsequent short-form references. This is useful where the citation must include some explanatory note on its first occurrence. For example:

```
\newinbook{rajarole}{Y.A.M. Raja Azlan
Shah^ (as he then was, being later
elected {\emYang di-Pertuan
Agong\})^}{The Role of Constitutional
Rulers in Malaysia}{\chapter 5 of\
Constitution of Malaysia: Further
Perspectives and Developments}{F.A.
Trindade \& H.P. Lee, eds. 1986}8
```

In this example, the explanatory note of the author's personal title will appear only in the first occurrence of the citation.

In optional page and section number references appended using `+`, `-`, `++` and `+-`, both `^...^` and `..._` can be used. In this context, any text within `^...^` will appear only in the text, and not in the tables, while `..._` conversely suppresses enclosed material in the text, but does export it to the table. This is useful for limiting the level of detail in statutory references within a table of statutes. A complex example might run as follows:

<sup>7</sup> Please note that this may cause conflicts with other styles. The next release should offer the option of using ordinary control strings for this purpose, rather than active characters.

<sup>8</sup> This expands in the text to: Y.A.M. Raja Azlan Shah (as he then was, being later elected *Yang di-Pertuan Agong*), *The Role of Constitutional Rulers in Malaysia*, CHAPTER 5 OF THE CONSTITUTION OF MALAYSIA: FURTHER PERSPECTIVES AND DEVELOPMENTS (F.A. Trindade & H.P. Lee, eds. 1986)

```
\constsch+-{5, para^s^..~5} \& {_5,
para..~8}9
```

If a table is produced from this citation, it will contain two entries:

```
\statutetableentry{{constsch}}{\rm
\ltoke specialface ={\sc }Federal
Constitution}{\rm \ltoke specialface =
{\sc }}!{\ sch..~5, para..~5}}{1006}
\statutetableentry{{constsch}}{\rm
\ltoke specialface ={\sc }Federal
Constitution}{\rm \ltoke specialface =
{\sc }}!{\ sch..~5, para..~8}}{1006}
```

**Iterative file writes.** Citations to statutes often refer to more than one section. References to multiple page numbers of other types of works do not cause any problem, because these specific page references are ignored when the entry is written to the external table. But for statutes, each section referred to becomes a sub-item in the statute table. It would not do to have a string of sections written as a single entry.

If the user enters the punctuation between section numbers as `\,`, `\&` or `\dash`, rather than simply `,`, `&` or `--`, each section reference will be separately written to the statute table file, as illustrated above. Note, however, that `makeindx` performs an alphabetic, not a numeric sort of the sections; you may need to shift the order of entries around by hand before the `*.sta` table is finally printed.

## Further Work

In its current form, L<sup>A</sup>T<sub>E</sub>X has realized most of the goals of its author. Citations can be defined with a wide variety of typeface combinations. Subsequent abbreviation is sensitive both to context and to the type of citation. Page and section numbers are simple to enter, and are correctly integrated into the final citation. Bibliographic tables can be produced automatically, and multiple statutory section numbers are easily entered and correctly parsed for entry into the table of statutes. Alas, there is always room for improvement.

**Memory requirements.** Because L<sup>A</sup>T<sub>E</sub>X stores all the information required to produce each citation in T<sub>E</sub>X's memory as the document is processed, it has a way of running out of space on large documents. A large T<sub>E</sub>X solves the problem for any document of reasonable length, but the style could be made less memory-hungry. The most obvious area of redundancy is in the paired `<blah>` and

<sup>9</sup> This expands in the text to: Federal Constitution sch. 5, paras. 5 & 8.

\<blah>full macro definitions. Most of the information stored in both of these macros could be hived off into the master storage macro for that citation name, say \@<blah>. Then this storage macro can be recalled by either of the citation nickname macros, which would then only need to add or alter any bits and pieces necessary to their own purposes.

The design of L<sup>A</sup>T<sub>E</sub>X emerged as its author struggled to learn the basics of T<sub>E</sub>X programming; it is therefore a certainty that there is room for memory savings and optimization in other parts of the code as well.

**Emacs Lisp support.** The Lisp extension language for the Emacs editor is powerful enough to provide context-sensitive support for L<sup>A</sup>T<sub>E</sub>X itself. That is, prompting for command completion and arguments could be made dependent upon the contents of the file `lexicite.tex` and the citation template declarations made to the current point in the document.

I am not a Lisp programmer myself, but the essentials of such a set of routines would be:

1. The ability to scan the `lexicite.tex` file and, for each citation style, identify:
  - The style name;
  - The number of arguments required by the style; and
  - Which elements are used, and which ignored, by the style.
2. The ability to scan the region from the current point to the beginning of the document (including any sub-documents), and identify the nicknames of citations defined, noting the style associated with them.
3. The ability to offer to define a new citation using an existing citation template. In proper Emacs fashion, the `[TAB]` key should perform auto-completion, or offer a list of available citation templates consistent with a partial entry. Once a template is selected, a prompt in the Emacs mini-buffer should call for each element of the citation as required.
4. The ability to provide auto-completion or a list of any citation nickname defined in the region between the current point and the beginning of the document (including sub-documents).

For novice users, this would ensure that citations are entered correctly, whatever package of blueprints is contained in `lexicite.tex`. In very long documents, it would make it easy to ensure that, in revision, nicknames are never used before they are initially defined.

**B<sub>B</sub>T<sub>E</sub>X analog.** Some users may prefer to maintain their citations in a separate file, as is done with B<sub>B</sub>T<sub>E</sub>X. With a little re-drafting, a similar approach could be adopted by L<sup>A</sup>T<sub>E</sub>X, without abandoning the ability to enter citations into the document directly. Two primary changes would be required. One is an environment macro (say, `\stashlcites`) which would perform the storage operation silently (without introducing citations into the text or the tables). The other is to introduce a toggle for each nickname macro, which would indicate whether it has once been cited in the document, so that a full-form citation is produced the first time, and short-form citations thereafter.

**Proxies.** Some sources have long and cumbersome names. Both good taste and clarity demand that these be replaced by a shorter form in subsequent references. Foreign statute names are also often customarily referred to by an abbreviation that cannot be derived directly from the text units of a L<sup>A</sup>T<sub>E</sub>X citation. To cover these cases, L<sup>A</sup>T<sub>E</sub>X should provide one more option (probably `=`), whose usage would follow that of `+`, `++`, `+-`, `*` and `-`. This nickname needs to be properly incorporated into full citations, and should replace the standard text of subsequent citations wherever appropriate.

**Parallel citations.** The most difficult problem still to be addressed is that of parallel citations. Law cases, in particular, often appear in more than one source. Some journals require that the specific page reference for a proposition be given for *each* source, even in subsequent abbreviated citations.

The problem is clear. In its current form, L<sup>A</sup>T<sub>E</sub>X only knows about the six memorized citation elements, plus a single page reference. Parallel citations require that L<sup>A</sup>T<sub>E</sub>X memorize the complete details for the remaining portions of the citation, and accept additional specific page numbers if necessary.

The syntax and programming strategy for this is in hand, but I will hold radio silence until I have a working solution to release.

### Page Ranges

George Greenwade has suggested that the page specifier set in initial citations should contain an option for specifying a range, which would then be enforced in subsequent citations. Because the accuracy of legal documents is often of critical importance, this is clearly a good idea, and will be included in the next release of L<sup>A</sup>T<sub>E</sub>X.

## Ending Note

LEX<sub>T</sub>E<sub>X</sub> is free of charge. The only conditions attached to its use are, first, that it be circulated as a complete package, together with its copyright and permission notices, and second, that LEX<sub>T</sub>E<sub>X</sub> and its author be duly cited whenever it is used in the preparation or production of material for publication. I am always open to bug reports, and happy to receive suggestions for improvement.

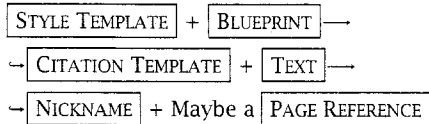
I hope that LEX<sub>T</sub>E<sub>X</sub> will prove useful enough to attract more law journals and legal publishers into the T<sub>E</sub>X fold. In particular, many law journals today would benefit from T<sub>E</sub>X's proficiency in typesetting foreign scripts. If this ability is combined with greater speed and certainty in the preparation of legal manuscripts, perhaps we shall see the alternatives rejected for failure to state a claim.

### Standard LEX<sub>T</sub>E<sub>X</sub> ver. 1.8 Citation Templates

```
\newbook{blah1}{B. Cardozo}{The Growth of The Law}{1924}+{15}.
→B. CARDOZO, THE GROWTH OF THE LAW 15 (1924).
\newbook{blah2}{}{Eastern Air Lines, Inc., 1978 Annual Report}{1979}+{15}.
→EASTERN AIR LINES, INC., 1978 ANNUAL REPORT 15 (1979).
\newman{blah3}{Comics Magazine Ass'n of America}{Press Release No.~51}{Sept.~16, 1954}+{15}.
→Comics Magazine Ass'n of America, Press Release No. 51 15 (Sept. 16, 1954).
\newman{blah4}{}{Telephone conversation with Douglas Borisky, Senior Revising Editor of the
\Columbia Law Review\}{Apr. 10, 1986}.
→Telephone conversation with Douglas Borisky, Senior Revising Editor of the Columbia Law Review (Apr. 10, 1986).
\newarticle{blah5}{Cox}{Federalism and Individual Rights}{73~Nw.~U.L. Rev.}{1}{1978}+{15}.
→Cox, Federalism and Individual Rights, 73 NW. U.L. REV. 1, 15 (1978).
\newinbook{blah6}{\O.W. Holmes}{Law in Science and Science in Law}{\in\ Collected Legal
Papers}{210}{1920}.
→O.W. HOLMES, Law in Science and Science in Law, IN COLLECTED LEGAL PAPERS 210 (1920).
\newinbook{blah7}{Maitland}{The Mystery of Seisin}{\in\ 3 Select Essays in Anglo-American
Legal History}{591}{1909}.
→Maitland, The Mystery of Seisin, IN 3 SELECT ESSAYS IN ANGLO-AMERICAN LEGAL HISTORY 591 (1909).
\newnews{blah8}{}{Abscam Jury Sees Videotape of Deal}{San Francisco Chron., Aug.~14, 1980}
{14, col.~1}{}.
→Abscam Jury Sees Videotape of Deal, San Francisco Chron., Aug. 14, 1980, p. 14, col. 1.
\newcase{blah9}{Baker v. \ Fortney}{299~S.W.2d}{563}{Mo. \ Ct. \ App. \ 1957}+{564}.
→Baker v. Fortney, 299 S.W.2d 563, 564 (Mo. Ct. App. 1957).
\newcase{blah10}{Clough Mill Ltd v Martin}{[1984] 3~All ER}{982}+{986}.
→Clough Mill Ltd v Martin, [1984] 3 All ER 982, 986.
\newjcase{blah11}{Decision of the Tokyo District Court, February 18,
1988}{Hanrei jih\=o, n.~1295}
{132}+{133}.
→Decision of the Tokyo District Court, February 18, 1988, Hanrei jihō, n. 1295 p. 132, 133.
\newstatute{blah12}{Robinson-Patman Act}{15~U.S.C.}{1982}+{{13^--13b^}\,
{21^a^}}.
→Robinson-Patman Act, 15 U.S.C. s. 13-13b, 21a (1982).
\newconstitutionart{blah13}{Federal Constitution}{1957}+{1}.
→Federal Constitution art. 1 (1957).
\newconstitutionsch{blah14}{Federal Constitution}{1957}+{1}.
→Federal Constitution sch. 1 (1957).
```



### LEX<sub>TEX</sub><sup>a</sup> Reference Card



After a citation template declaration for a particular work, use the nickname to refer to that source thereafter.

The style template is called once, by `lexitex.sty` at startup. Blueprints for use by the style template are contained in `lexicite.tex`. The distribution contains blueprints for the style template which will suit most, but by no means all, purposes.

#### Options and Special Commands

`\lowercaseblah` before any LEX<sub>TEX</sub> citation, forces *Id.* to lowercase, should it occur.

`\lexicite{blah1}` is used to execute nicknames that include a number, such as a date.

`\...\` marks special typeface (usually roman) within an argument of a citation template declaration.

`\lexiproof{blah}` is used to generate a proof sheet giving most of the permutations of a nickname fired by a given citation template. This is useful when creating or editing citation templates.

\* used immediately after a LEX<sub>TEX</sub> citation, suppresses output of a bare citation template declaration or nickname.

- used between a LEX<sub>TEX</sub> citation and an optional page or section reference, suppresses output of the citation template declaration or nickname, and of the page reference.

<sup>a</sup> LEX<sub>TEX</sub> is by Frank G. Bennett, Jr., Lecturer in Law at the School of Oriental and African Studies, University of London.  
(Internet) fbennett@clus1.ulcc.ac.uk

- + appends a page or section number when typed after a citation template declaration or a nickname. Always enclose in braces: `+{123}`.
- ++ appends a plural page or section argument to any LEX<sub>TEX</sub> citation. Enclose page numbers in braces, and separate items with `\,`, `\&` or `\dash` (ranges not supported, but do no harm when used with non-statute citation templates and nicknames).
- +− appends a singular page or section argument, but writes multiple references to the table file. Most often used in combination with `^...` and `...` to cope with references to multiple subsections of a single statutory provision.
- ^ used within appended page or section references to enclose material that will appear only in the text, not in the table file. Used in all other arguments to enclose material that will appear only in initial or full-form citations.
- − used within appended page or section references to enclose material that will appear only in the table file, not in the text.

#### Tables

To write external table files, use:

- `\makebooktable` (makes \*.btb)
- `\makearticletable` (makes \*.atb)
- `\makecasetable` (makes \*.ctb)
- `\makestatutetable` (makes \*.stb)

Process the external table file using `makeindx` with the appropriate style file and output file name:

- `lexibook.ist − *.bok`
- `lexiarti.ist − *.art`
- `lexicase.ist − *.cas`
- `lexistat.ist − *.sta`

Place finished tables in the document using:

- `\printbooktable`
- `\printarticletable`
- `\printcasetable`
- `\printstatutetable`