Full, Short, and *Id.* Citations (`state.dtx`)

Charles Duan

May 2, 2023

Except in limited circumstances,[1] this package processes citations linearly starting from the beginning of the document and never looking backwards or forwards. As a result, it must constantly store "state" information about citations previously seen, in order to generate contextually correct citations.

Two types of state information are discussed here: information about where references were first or previously cited, and information about the last citation shown.

1 Short Forms

For most references, there is a full citation form and a short one. The full form is used first, and the short one used subsequently according to one of two rules: Some references use a *supra* short form that may be used subsequently throughout, while for others the short form may only be used if a full-form citation appeared in close proximity. The purpose of the previous citation state memory, described below, is to determine which of these two forms to use.

Complicating matters, citations can appear in footnotes and in body text, in both law review articles and legal memoranda, and both in inline and non-inline forms. All of these factors interact to affect whether a full or short form citation should be used.

This package follows the following rules to determine long and short forms:

1. An inline short form may be used in text after the full form is used in text, either inline or in a citation.
2. An inline short form may be used in a footnote (a) if an inline short form may be used in text, (b) after a full citation in this or any previous footnote, or (c) after a full inline form in this footnote.
3. A short cite may be used in text after the full citation is used in text.
4. A short cite may be used in a footnote (a) after the full citation has been used in text, (b) after the full citation has been used in a footnote if this is

[1]These are namely internal cross-references and Table of Authorities generation, described in `xref.dtx` and `toa.dtx` respectively, both of which use LaTeX's aux file system.

a supra-type short cite, or (c) if the full citation has been used in the last five footnotes.

\resetcitationforms

The command `\resetcitationforms` resets the aforementioned citation state, forcing all citations to be the full form.

Recordation of all full cites.

```
\def\hi@allfullcites{}
\def\hi@addfullcite#1{\gappto\hi@allfullcites{\let#1\relax}}
```

Resets all citation state.

```
\def\resetcitationforms{%
    \hi@allfullcites
    \gdef\hi@allfullcites{}%
    \def\do##1{%
        \cslet{@last@##1}\relax
    }%
    \hi@state@list
}
```

These rules are implemented as follows. For every reference, two citation state variables are maintained: `\dfc@`⟨*ref*⟩, and `\dfi@`⟨*ref*⟩. The former pertains to citations, and the latter to inline textual references.

For the `\dfi@`⟨*ref*⟩ forms, there are three possible values:

- `\relax` (undefined): The full inline form should be used in all situations.
- `\@ne` (1): The short form should be used in footnotes only. Where the short form should only be used in the current footnote (rule 2(c)), this state variable is set locally to the footnote.
- `\tw@` (2): The short form should be used in all situations.

For the `\dfc@`⟨*ref*⟩ forms:

- A value of zero indicates that the reference has been cited in text but no footnote.
- A positive value indicates that the reference has been cited only in a footnote. The value is the footnote number.
- A negative value indicates that the reference has been cited both in text and in a footnote. The absolute value is the footnote number.

Note that while `\dfi@`⟨*ref*⟩ values are number registers, the `\dfc@`⟨*ref*⟩ values are text numbers.

\ShowCitationState

The macro `\ShowCitationState{`⟨*ref*⟩`}` gives a textual account of the state of a reference at the point that the macro is called.

```
\def\ShowCitationState#1{%
    \immediate\write16{For citation reference `#1',
        \@ifundefined{dfi@#1}{the full inline form will be used}{%
            \ifcase\csname dfi@#1\endcsname
                an illegal inline state was encountered%
                \or the short inline form will be used in footnotes only%
                \or the short inline form will be used%
                \else an illegal inline state was encountered%
            \fi
        }, and
        \@ifundefined{dfc@#1}{the full citation form will be used}{%
            \ifnum\csname dfc@#1\endcsname>\z@
                the full citation was used in footnote
                \csname dfc@#1\endcsname
            \else
                the full citation was used in text so the short form will be
                used%
            \fi
        }.%
    }%
}
```

\hi@record@cite

This utility macro is run every time a reference is cited, as part of the citation drawing routines in `draw.dtx`. It updates state variables indicating whether the full or short citation form should be used. The macro runs `\hi@record@cite@inline` in all cases, and `\hi@record@cite@` only on non-inline citations.

I was formerly of the opinion that a full citation inside another one (e.g., inside a parenthetical) should have no effect on later use of the full citation. I'm now of a different opinion, because especially for lengthy full citations, it looks weird for there to be a full citation inside a parenthetical and an identical full citation immediately after. However, inline citations will not be affected by parentheticals.

```
\DeclareRobustCommand\hi@record@cite[1]{%
    \if@hi@in@toa\else
        \expandafter\hi@addfullcite\csname dfc@#1\endcsname
        \expandafter\hi@addfullcite\csname dfi@#1\endcsname
        \ifnum\hi@citelevel<\tw@
            \expandafter\hi@record@cite@inline\csname dfi@#1\endcsname
        \fi
        \if@hi@inline\else
            \expandafter\hi@record@cite@\csname dfc@#1\endcsname{#1}%
        \fi
    \fi
}
```

Records an inline citation, setting `\dfi@`⟨*ref*⟩ (which is #1) to the appropriate value.

```
\def\hi@record@cite@inline#1{%
    \ifx#1\relax \let#1\z@ \fi
    \ifnum#1<\tw@
        \if@hi@note
            \if@hi@inline\else \expandafter\global \fi
            \let#1\@ne
        \else
            \global\let#1\tw@
        \fi
    \fi
}
```

Records a citation. #1 is the `\dfc@`⟨*ref*⟩ macro, and #2 is the reference name.

```
\def\hi@record@cite@#1#2{%
```

If this is the first citation, record the footnote number or zero if in text.

```
    \ifx#1\relax
        \xdef#1{\if@hi@note \the\c@footnote \else 0\fi}%
    \else
        \if@hi@note
```

In a footnote, where this citation was previously used. If it was previously used only in text, set it to the negative of the footnote number. Otherwise, if this is a non-supra citation, update the state value to reflect the current footnote number.

```
            \ifnum#1=\z@
                \xdef#1{-\the\c@footnote}%
            \else
                \@ifundefined{sc@#2@supra}{%
                    \xdef#1{\ifnum#1<\z@ -\fi\the\c@footnote}%
                }{}%
            \fi
        \else
```

In text, where this citation was previously used. Just make sure the value is negative.

```
            \xdef#1{\ifnum#1>\z@ -\fi#1}%
        \fi
    \fi
}
```

`\hi@record@choose` Choose a full or short form for the given reference. This takes three parameters: the reference nickname, the callback for a full form, and the callback for a short form.

```
\def\hi@record@choose#1{%
    \@test \if@hi@inline \fi{%
        \hi@record@choose@inline{#1}%
    }{%
        \hi@record@choose@cite{#1}%
    }%
}
```

Choose a full or short inline form.

```
\DeclareRobustCommand\hi@record@choose@inline[1]{%
    \@ifundefined{dfi@#1}{\@firstoftwo}{%
        \@test \ifnum\csname dfi@#1\endcsname=\tw@ \fi{\@secondoftwo}{%
            \@test \if@hi@note \fi{\@secondoftwo}{\@firstoftwo}%
        }%
    }%
}
```

Choose a full or short cite form.

```
\DeclareRobustCommand\hi@record@choose@cite[1]{%
    \expandafter\hi@record@choose@cite@\csname dfc@#1\endcsname{#1}%
}
\def\hi@record@choose@cite@#1#2{%
    \@test \ifx#1\relax \fi{\@firstoftwo}{%
        \@test \if@hi@note \fi{%
```

Within a footnote, where the citation has previously been used. If the value is negative or zero (i.e., there was a citation in text), then always use the short form. If this is a supra-type citation, always use the short form. Otherwise, follow the five-footnote rule.

```
            \@test \ifnum #1>\z@ \fi{%
                \@ifundefined{sc@#2@supra}{%
                    \@tempcnta=\c@footnote
                    \advance\@tempcnta-5\relax
```

Run `\@firstoftwo` if the last citation number was less than the current footnote number minus five, and `\@secondoftwo` otherwise

```
                    \@test \ifnum #1<\@tempcnta \fi
                }{\@secondoftwo}%
            }{\@secondoftwo}%
        }{%
```

Within text. Run `\@firstoftwo` if positive (i.e., only in footnotes) and `\@secondoftwo` if negative or zero.

```
            \@test \ifnum#1>\z@ \fi
        }%
    }%
}
```

1(a) First Full Citation For references that follow the five-footnote rule, formatting sometimes will depend on whether a full citation is the first full citation in the document or a later repeated one. For example, consider a case citation where several cases with the same parties are being cited. Consider the following footnote citations:

1. Plaintiff v. Defendant (*Plaintiff I*), 123 F. Supp. 456 (S.D.N.Y. 1980).

10. Plaintiff v. Defendant (*Plaintiff II*), 234 F.2d 567 (2d Cir. 1981).

25. *Plaintiff I*, 123 F. Supp. 456 (S.D.N.Y. 1980).

27. *Plaintiff I*, 123 F. Supp. at 460.

The citation in footnote 25, despite being a full citation, ought to use the short-form case name established earlier. It would be odd to repeat the citation in footnote 1 entirely, because that would duplicatively include the short-form parenthetical. But omitting the parenthetical renders footnote 27 ambiguous. As a result, the best presentation for footnote 25 is to use just the established short-form name with the rest of the case fully cited.

This package accounts for this problem by recording whether a reference following the five-footnote rule has been cited previously, enabling citation macros to adjust the contents of full citations accordingly.

`\hi@record@firstfullcite` Determines if this citation has had at least one full citation. This will be true if `\dfc@`⟨*ref*⟩ is defined at all. It is useful for citations that follow the five-footnote rule, as it is sometimes relevant to know whether a full citation is the first full citation ever or a repeated full citation. Takes three arguments: #1 is the reference name, #2 is the callback for the first full citation ever, and #3 the callback otherwise.

```
\def\hi@record@firstfullcite#1{%
    \@ifundefined{dfc@#1}%
}
```

2 *Id.* Forms

In addition to long and short citation forms, a citation to a reference may be replaced with *id.* or otherwise altered if the immediately preceding citation contains duplicative information. Another set of citation state information must be maintained for this.

Generally, *id.* may be used and citation information may be omitted when it is the same as the previous citation. Assuming that `2 ref at 46` was previously cited, then:

Next input	Produces	Explanation
`2 ref at 46`	*Id.*	Volume and page omitted
`2 ref at 48`	*Id.* at 48	Volume omitted
`3 ref at 50`	3 *id.* at 50	Both numbers included
`3 ref at 46`	3 *id.* at 46	Page cannot be omitted, as volume differs

However, there are two main exceptions to this information omission process. First, not every duplicative element can be omitted, if a "higher level" element differs. In the last example above, the page number 46 is duplicative, but it cannot be omitted because the volume number differs. Second, the presence of multiple citation items in a context can prevent the use of *id.* In this example:

```
\sentence{see ref1 at 5; ref2 at 7}. Further text.
\sentence{see ref2 at 7}.
```

No *id.* may be used in the final citation, even though the reference and page are identical to the immediately preceding one, because the first citation string included two distinct references so *id.* would be ambiguous. However:

```
\sentence{see ref1 at 5; but see ref1 at 7}.
Further text. \sentence{see ref1 at 7}.
```

Here, the last citation should be formatted "*See id.* at 7." The *id.* is permissible because only `ref1` was cited in the first citation string, but the page number cannot be omitted. Furthermore, a footnote can use *id.* in its first citation only if the previous footnote cited only one reference.

To implement all of this, the package defines several state variables, generally reflecting elements of citation items. Each state variable maintains four pieces of information:

- A current value, set while the citation item is being processed and formatted
- A last value, reflecting the previous citation item's elements to current citation item is allowed to use those previous values

- A flag indicating whether the state variable has received too many values within a citation string such that it cannot be used for element omission in a subsequent string. This "state of the state variable" flag can take on one of three conditions:
  - Unset: The value at the start of the citation string.
  - Set: A citation item has set the value of the state variable, and it may be used in a future citation string.
  - Invalid: The state variable has been set two or more times to different values. It cannot be used in future citation strings, and it invalidates "lower-level" state variables as well.
- An "invalidation list," namely the list of lower-level state variables that become invalid if this state variable is invalid.

List of state variables.

```
\def\hi@state@list{}
```

Create a state variable. This (1) adds it to a list and (2) creates four macros for it: `\@this@`⟨*var*⟩, `\@last@`⟨*var*⟩, `\hi@sv@st@`⟨*var*⟩ (the state), and `\hi@sv@inv@`⟨*var*⟩ (the invalidation list).

```
\def\hi@state@var#1{%
    \expandafter\ifx\csname hi@sv@inv@#1\endcsname\relax\else
        \PackageError\hi@pkgname{%
            State variable #1 already defined.\MessageBreak
            Redefining it
        }{}%
    \fi
    \addto@macro\hi@state@list{\do{#1}}%
    \cslet{@this@#1}\relax
    \cslet{@last@#1}\relax
    \cslet{hi@sv@st@#1}\z@
    \cslet{hi@sv@inv@#1}\@empty
    \hi@state@setinv{#1}{#1}%
}
```

Add a state variable #2 that will be invalidated when state variable #1 changes.

```
\def\hi@state@setinv#1#2{%
    \edef\reserved@a{%
        \noexpand\let
        \expandafter\noexpand\csname hi@sv@st@#2\endcsname
        \noexpand\tw@
    }%
    \expandafter\add@macro@to@macro \csname hi@sv@inv@#1\endcsname\reserved@a
}
```

What follows is the list of state variables, which is used to generate a table in the documentation. Please consult the source code for the actual list.

2.1 List of State Variables

Variable	Invalidates
case	page, orig@page, inline, opt, vol, title
orig@page	
page	orig@page
inline	
opt	
vol	title, page, orig@page, opt
title	vol, page, orig@page, opt

2.2 Updating Citation State

The citation state needs to be updated for each citation string, citation item, and footnote. The hook macros defined below are used throughout `draw.dtx`.

At the beginning of a citation string, all the variables' last values are based on the state of the system prior to the citation, and they all have a condition of Unset.

```
\def\hi@state@startstring{%
    \def\do##1{\cslet{hi@sv@st@##1}\z@}\hi@state@list
}
```

At the beginning of each citation element, all current values are set to `\relax`. Each element in a citation string can rely on the last values in displaying itself, and can set the current variables as desired. Some current variables may be set prior to the citation-specific macros (e.g., `\@this@case`).

```
\def\hi@state@startelt{%
    \def\do##1{\cslet{@this@##1}\relax}\hi@state@list
}
```

At the end of each element in the citation string, all the current values are checked against their last values.

- If the condition is Unset then the condition becomes Set.
- If the condition is Set and the last value equals the current value, do nothing.
- Otherwise, the condition becomes Invalid, and the condition for any dependent state variables become Invalid.

In all cases, the last value becomes the current value, because within a single citation string the invalidity state is irrelevant.

```
\def\hi@state@endelt{%
    \def\do##1{%
        \expandafter\ifcase\csname hi@sv@st@##1\endcsname
            % Unset
            \cslet{hi@sv@st@##1}\@ne
        \or
            % Set
            \hi@state@test{##1}{}{\csname hi@sv@inv@##1\endcsname}%
        \fi
        \csletcs{@last@##1}{@this@##1}%
    }\hi@state@list
}
\def\hi@state@test#1{%
    \@expand{\expandafter\hi@state@test@\csname @this@#1\endcsname}{%
        \csname @last@#1\endcsname
    }{i}%
}
\def\hi@state@test@#1#2{%
    \ifx#1#2\expandafter\@firstoftwo\else\expandafter\@secondoftwo\fi
}
```

At the end of a citation string, all Invalid variables have their last value set to `\relax`, and all Set variables retain their last value. (There should be no Unset variables since every variable has its state reviewed for each citation element, and there must be at least one citation element.)

```
\def\hi@state@endstring{%
    \def\do##1{%
        \expandafter\ifnum\csname hi@sv@st@##1\endcsname=\tw@
            \cslet{@last@##1}\relax
        \fi
        \cslet{@this@##1}\relax % Unnecessary but prophylactic
    }\hi@state@list
}
```

At the end of a footnote, the last values of the state variables are either globalized or relaxed, depending on whether there was one or more references cited.

```
\def\hi@state@endnote{%
    \ifnum\hi@note@cites=\@ne
        \def\do##1{%
            \expandafter\hi@state@endnote@\csname @last@##1\endcsname
        }\hi@state@list
    \else
        \def\do##1{%
            \global\expandafter\let\csname @last@##1\endcsname\relax
        }\hi@state@list
    \fi
}
\def\hi@state@endnote@#1{%
    \global\let#1#1%
}
```

For inline citations, special handling of citation state is required as described in `iface.dtx`. Thus, several macros are provided for saving all the state variables to a macro so that they can be restored or reset later.

Saves all the `\@last@`⟨*ref*⟩ state variables to a macro. #1 is the macro.

```
\def\hi@state@save#1{%
    \let#1\@empty
    \def\do##1{%
        \expandafter\hi@state@save@one\csname @last@##1\endcsname#1%
    }\hi@state@list
}
\def\hi@state@save@one#1#2{%
    \ifx#1\relax
        \addto@macro#2{\let#1\relax}%
    \else
        \addto@macro#2{\def#1}%
        \expandafter\addto@macro\expandafter#2\expandafter{%
            \expandafter{#1}%
        }%
    \fi
}
```

Resets all last state variables.

```
\def\hi@state@reset{%
    \def\do##1{\expandafter\let\csname @last@##1\endcsname\relax}\hi@state@list
}
```

3 Footnote State Management

Footnotes interact closely with both short and *id.* citations: Whether a short citation can be used depends in part on whether the full citation was in text or a footnote, and *id.* citations can be used at the start of a footnote depending on the content of the previous footnote. How footnotes interact with citation forms also depends in part on whether a document is a law review or legal memorandum document.

As a result, several accounting tasks must be performed at the beginning and ending of footnotes. For all types of articles, a flag is set indicating that text is in a footnote, and at the end of footnotes any deferred note text that has not been emitted (as described in `parens.dtx`) is put at the end of the note.

For law review articles, a counter is kept of how many unique citations have appeared in the footnote. Each reference cited advances the counter by one, except *id.* citations advance the counter only from zero to one. The *id.* system above uses this counter to determine state after the footnote. Legal memoranda, on the other hand, prohibit *id.* cites at the beginnings of footnotes, to avoid ambiguity as to whether *id.* refers to the last footnote or the last in-text citation.

```
\AtBeginDocument{%
    \let\hi@footnotetext\@footnotetext
    \long\def\@footnotetext#1{%
        \hi@footnotetext{\hi@footnote@pre#1\hi@footnote@post}%
    }%
}%
\newcount\hi@note@cites
\appto\hi@hooks@review{
    \def\hi@note@advance{\advance\hi@note@cites\@ne}
    \def\hi@note@advance@id{\ifnum\hi@note@cites=\z@\hi@note@advance\fi}
    \def\hi@footnote@pre{%
        \@hi@notetrue
        \hi@note@cites\z@
    }
    \def\hi@footnote@post{%
        \hi@deferred@note@endnote
        \hi@state@endnote
        \global\let\@last@inline\relax
    }
}
\appto\hi@hooks@memo{
    \let\hi@note@advance\@empty
    \let\hi@note@advance@id\@empty
    \def\hi@footnote@pre{%
        \@hi@notetrue
        \hi@noid
    }%
    \def\hi@footnote@post{%
        \hi@deferred@note@endnote
    }%
}
```

Conditional for determining when we're inside a note.

```
\newif\if@hi@note \@hi@notefalse
```