State secrets in bibliography-style hacking

Karl Berry and Oren Patashnik

BibTeX output, in essence, consists of chunks of information separated by punctuation. The information chunks are things like the formatted author's name, the title of the work being cited, or its publication date. And the punctuation separator between chunks, in BibTeX's standard styles, is either a comma or a period.

But exactly how the bibliography styles (`.bst` files) determine and then output that separating punctuation, for even moderately experienced .bst-file hackers, is somewhat of a mystery, or even a complete secret. This short article demystifies the process, and exposes a secret.

When the bibliography-style output routines are ready to send a chunk of information to the output (`.bbl`) file, they generally don't know what punctuation follows that chunk; it might be a comma or it might be a period, depending on which chunk of information comes next. So the output routines (for BibTeX's standard styles, and for many others) keep the previous information chunk on its working stack, and when the current chunk is ready for output — at which point the correct separator between the previous and current chunks is known — they output that previous chunk, along with the now-known separator, to the `.bbl` file, leaving the current chunk on the stack until the next chunk is ready for output. And the process repeats.

And how do the output routines know what separator to use?

The answer is in a comment in `btxbst.doc`, BibTeX's documentation (and template) file for the standard styles (all code blocks here are reformatted for *TUGboat*):

```
% To tell which separator is needed,
% we maintain an output.state.
```

(Aside: Much of the reason for the mystery/secret is historical. Originally that documentation line from `btxbst.doc` was stripped when the standard `.bst` files were generated. Over the years, other styles — for example, those generated from `makebst` — similarly did not contain that documentation line. Thus it remained a sort of secret, hidden away in `btxbst.doc`.)

So it's the `output.state` variable that determines between-chunk punctuation.

And how does a style hacker who wants to introduce a new punctuation mark implement that?

Answer: Create a new output state.

A recent IEEE-like bibliography style shows how that's done. That style wanted to separate any url (which normally appears at the end of the entry) from previous information with a semicolon.

An example formatted entry:

[1] H. Kopka and P.W. Daly, *Guide to LaTeX*, Addison-Wesley Professional, 4th edition, 2003; `amazon.com/dp/0321173856`.

It's the semicolon before the url that's new. Here's a description of the fairly straightforward pieces of new code that `ieeelike.bst` uses to implement the new style. (Recall that the `.bst` language is (mostly) stack-based and uses postfix notation, something like PostScript.)

First there's code to create a new output state, say with the symbolic name `after.url.separator`, so-named because the url chunk that's being created comes after the semicolon that separates it from the previous chunk.

```
INTEGERS {
      output.state before.all mid.sentence ...
      after.url.separator }   % new output state
...
#4 'after.url.separator :=  % unique state value
```

Next, code to change the current output state to our new one when the url chunk is created:

```
FUNCTION {url.block}
{       % change the output state appropriately:
  output.state before.all =
    'skip$     % if entry starts with url (rare)
    { after.url.separator 'output.state := }
  if$
}
...
FUNCTION {format.url}
{ ...
  url.block
  ⟨code to format the url chunk⟩
}
```

Finally, the output-routine code that, for the desired output state, writes out the previous chunk appended with the semicolon separator that will precede the url:

```
    { output.state after.url.separator =
        { "; " * write$ }
```

See `ieeelike.bst` (linked from the *TUGboat* contents page for this issue) for the full context.

And that, in a nutshell, is how to add the new semicolon separator: by creating a new output state. The `output.state` secret has been leaked.

◇ Karl Berry and Oren Patashnik
https://tug.org/bibtex