The `poetry` Package, v1.0

Donald P. Goodman III

June 19, 2017

Abstract

The `poetry` package provides some macros and general doodads for type-setting poetry. There is, of course, already the excellent `verse` package, and the `poetrytex` package provides some extra functionality on top of it. But `poetry` provides much of the same functionality in a bit of a different way, and with a few additional abilities, such as facilities for a list of poems, an index of first lines, and some structural commands.

Contents

1   Introduction

The `verse` package provides some truly excellent verse typesetting abilities. Its documentation, particularly, is a great demonstration of (1) how to write LaTeX documentation, and (2) how difficult typesetting poetry with LaTeX was before `verse` came along. And yet, occasionally one comes to some things that it doesn't seem able to do. At least, I personally did; when lines wrapped, the line number was placed along the *bottom* of the line, rather than the top, and there doesn't seem to be any way to fix this rather odd behavior.

So, the `poetry` package was born. It endeavors to provide all the same functionality as `verse`, in a (mostly) compatible way; but also to fill in this little lacuna, plus any others I happened across while doing it. It also endeavors to add similar functionality as I found in the other major LaTeX poetry package, `poetrytex`, which includes some great ideas for document structure. Finally, it adds some new functionality which is not, to my knowledge, included elsewhere.

This document is written in accordance with the LaTeX DOCSTRIP utility, which allows the automatic extraction of package code and documentation.

The numbering in this document is in *dozenal*; that is, numbering goes 1, 2, 3, 4, 5, 6, 7, 8, 9, ᘔ, Ɛ, 10, 11 . . . It uses the `dozenal` LaTeX package to make this happen. For more information, visit http://www.dozenal.org.

2   Typesetting Verse

2.1   Basics

poem

\poem

`poetry`'s basic environment is, as one would expect, the `poem` environment. `\begin{poem}` and `\end{poem}` will get your poem off and running, no questions asked. The environment redefines certain commands, particularly `\\`, so be aware that you can't use `\\` just anywhere within your poem. Alternatively, you can use `\poem` to begin the poem, and just use the normal `\\-` to end it (about which more in a moment).

Conceptually, each line of the poem is a paragraph, so in order to ensure proper paragraph setting, the `poem` environment also resets `\parindent`, `\hangindent`, and `\hangafter` to appropriate values. We will see how to customize those values in Section 2.2.

`\\`   End each line of your poem with `\\`, which will end the line and also number the next line. By default, every line is numbered; if you want some other behavior, you will find how to customize line numbering in Section 2.3.

`\\!`   If the poem has stanzas, end them with the command `\\!`; this will insert the appropriate amount of vertical spacing, as well as reset the verse line number counter.

`\\-`   To end the entire poem, after the last line use `\\-` (double backslash followed by a hyphen). This clearly differentiates normal line endings; line endings which also end stanzas; and the final line of the poem.

`\\*`   If you wish to end the poem for a moment, then resume it later, use `\\*`. You

can then insert whatever you want (e.g., a subpart heading, or some annotations).

* When you want to start the poetry again, begin the line with `\*`. This allows you insert mostly arbitrary material without leaving the `\poem` environment.

Finally, if you want to end the poem for a moment (that is, insert some non-poetical material, like a subpart heading or some annotations), but also want to

\\? end a verse, you can use `\\?`. This combines `\\!` and `\\*`; it has the same visual effect as the latter, but the semantic meaning of the former. Since non-poetical material will follow, it does *not* insert the inter-stanza spacing. You must still use `\*` when you restart poetical material.

The idea behind these commands is not to make life difficult for the typesetter, but to mark semantic concepts; that is, to mark very clearly when a line ends; when a line ending is also a stanza ending; when a line ending is followed by non-poetry material; and when a line ending is also the final line of the poem. This kind of information makes it much easier for format converters to work, as well as for semantic readers to figure out what's going on.

If you need to include some non-poetry material right at the beginning of a

\poetry* poem, use the starred version to begin it: `\poetry*` or `\begin{poem}*`. This will
\begin{poem}* suppress the first line number. When you're ready to begin your first line, begin it with `\*` to insert the first line number.

\hin If a line needs some extra indentation, indent it with `\hin`. This adds the space `\poemhinwd` to the left margin of that particular line and no other. All other paragraph settings are adjusted accordingly. This command can be repeated as needed to produce specific indentation.

An example of this basic usage:

```
\begin{poem}
Let me not to the marriage of true minds \\
\hin Admit impediments.  Love is not love \\
Which alters when it alteration finds, \\
\hin Or bends with the remover to remove: \\!

O, no!  it is an ever-fixed mark, \\
\hin That looks on tempests and is never shaken; \\
It is the star to every wandering bark, \\
\hin Whose worth's unknown, although his height be taken. \\-
\end{poem}
```

Which produces (with drastically reduced line length, to show how wrapping will work in these cases; though keep in mind that all these lengths are customizable):

1 Let me not to the marriage of true
  minds

2 Admit impediments. Love is
   not love

3 Which alters when it alteration

<div style="text-align:center">finds,</div>

4 Or bends with the remover to
     remove:

5 O, no! it is an ever-fixed mark,
6   That looks on tempests and is
     never shaken;
7 It is the star to every wandering
     bark,
8   Whose worth's unknown, al-
     though his height be taken.

2.2 Customizing Spacing and Indentation

`poetry` inserts some space above and below the typeset poems (unless you tell it not to). `\poemtopskip` is inserted above the poem, and `\poembotskip` is inserted below it. These are standard LaTeX lengths, and can be reset in the usual way. By default, both are set at `1em`.

 `\poemindent` is the length that governs how far the poem will be indented from the left margin. Reset it as usual with LaTeX's `\setlength` command; e.g., `\setlength{\poemindent}{1.5in}` to indent the entire poem 1.5in from the left line. By default, `\poemindent` is `0pt`, meaning that verses are flush with the left margin. This also means that line numbers, if they are on the left, will hang into the left margin.

 It may be that you wish to control the overall widths of lines; for example, you may want poetry lines to be shorter than the actual text lines. For this, use `\poemmaxlinewd`. By default, this standard LaTeX length is set to equal `\linewidth`, so poetry lines can equal the length of the text block. If you don't want this, then you can set it as you please. E.g., if your text block is `5in` wide and you want a half inch of space on either side of your poetry lines, set `\poemmaxlinewd` to `4in` and `\poemindent` to `0.5in`.

 When lines wrap, `poetry` applies a certain `\hangindent` to make them look nice. To change the length of this hanging indent, reset the value of the standard LaTeX length `\poemhangindent`, in any of the usual ways (e.g., `\setlength`). By default, `\poemhangindent` is `1em`.

 Often, particularly when a poem (or lyrics) are simply a small portion of a largely prosaic work, we want the lines more-or-less centered within the text block. The `verse` package does this by providing an optional argument to the `verse` environment, giving the length of a line which will serve as the basis for centering the entire poem. `poetry` instead provides the `\centerpoem` commands. `\centerpoemon` sets the centering mechanism on, and (obviously) `\centerpoemoff` turns it off. `\centerpoemon` takes a mandatory argument, the line which you wish to serve as the basis for the centering:

<div style="text-align:center">`\centerpoemon` {⟨<i>line-to-use-for-centering</i>⟩}</div>

What this command does is measures the length of the line, compares it to the text block, and then centers that line; all other lines in the poem will share a left and right margin with that line. E.g.:

```
\centerpoemon{The ship has weather'd every rack, the prize we sought is won}
\begin{poem}
O captain! my captain!  The fearful trip is done; \\
The ship has weather'd every rack, the prize we sought is won;\\-
\end{poem}
```

Which gives us:

    1   O captain! my captain! The fearful trip is done;
    2   The ship has weather'd every rack, the prize we sought is won;

Note that this *does* set the maximum line width to the width of that line, as well. You can reset this after issuing \centerpoemon if that's not the behavior you want; that is, if you want to center on that particular line but still allow other lines to be longer. This behavior makes it default that your poem will actually be centered on the width of the line you've entered; however, if you're really just looking for an approximate optical centering, with lines extending longer if necessary, set \poemmaxlinewd to what you want your longest line to be. Just make sure that you set it *after* you set \centerpoemon.

\centerpoemoff simply resets \poemindent to 0pt and \poemmaxlinewd to \textwidth.

\poemhinwd   The indentation given by \hin is controlled by the LaTeX length \poemhinwd, which can be set however the user wishes. By default, it is 1em. E.g.:

```
\centerpoemon{Alas, poor Yorick!  I knew him.  Horatio!}
\settowidth{\poemhinwd}{Alas, poor Yorick!  I knew him.  }
\begin{poem}
Alas, poor Yorick!  I knew him.\\
\hin Horatio! \\
\setlength{\poemhinwd}{2em}
A fellow of infinite jest, of most excellent fancy.  \\
\hin He hath borne me on his back a thousand times, \\
and now, how abhorred in my imagination it is! \\
My gorge rises at it.  Here hung those lips\xelip\\-
\end{poem}
```

This is a common requirement when the meter of one line is continued onto the next, as in this famous line(s) from *Hamlet*:

    1   Alas, poor Yorick! I knew him.
    2   Horatio!
    3   A fellow of infinite jest, of most excellent
            fancy.

<div align="center">

4   He hath borne me on his back a
           thousand times,

5   and now, how abhorred in my imagina-
           tion it is!

6   My gorge rises at it. Here hung those
           lips . . .

</div>

This way we can advance the second line so that it's clearly a continuation of the first. The second `\hin` command here is just to show that the lengths can be adjusted at will throughout the poem, and it doesn't effect the line numbering in any way.

Using `\hin` manually works fine, but sometimes it's a hassle, particularly when the indentation pattern is regular. E.g., when every second line is to be indented, why should you manually have to insert `\hin` every second line? That's the sort of thing computers are good at. Fortunately, `poetry` agrees with you, and offers the `poemindentevery` counter. This is a standard LaTeX counter. If its value is 0, it has no effect whatsoever; if its value is any other integer, however, it will indent every multiple of that line number. E.g.:

<div style="margin-left:2em; font-style:italic; color:gray;">poemindentevery</div>

```
\setcounter{poemindentevery}{3}
\setcounter{poemlinenumsevery}{2}
\begin{poem}
Once upon a midnight dreary, while I pondered, weak and weary,\\
Over many a quaint and curious volume of forgotten lore---\\
While I nodded, nearly napping, suddenly there came a tapping,\\
As of someone gently rapping, rapping at my chamber door.\\
''`Tis some visitor,'' I muttered, ``tapping at my chamber door---\\
Only this and nothing more.''\\-
\end{poem}
```

Which will give us this:

<div align="center">

       Once upon a midnight dreary, while I
           pondered, weak and weary,

2   Over many a quaint and curious volume
           of forgotten lore—
       While I nodded, nearly napping, sud-
           denly there came a tapping,

4   As of someone gently rapping, rapping at
           my chamber door.
       "'Tis some visitor," I muttered, "tapping
           at my chamber door—

6   Only this and nothing more."

</div>

Notice that line numbers are also unaffected by this setting; line numbers will be placed where needed, and indents will be added as needed, without regard for

one another. (We will officially meet the counter `poemlinenumsevery` in Section 2.3.1.)

Sometimes, however, indenting every third line (or whatever) isn't the right solution; sometimes each *verse* has a particular indentation pattern. For that, `\poemvsindentlines`   poetry offers the command `\poemvsindentlines`. Redefine this command to contain a list of the line numbers (by verse) you wish to indent, and poetry will dutifully indent them. E.g.:

```
\def\poemvsindentlines{23}%
\begin{poem}
I saw you starting for another war,\\
The emblem of adventure and of youth,\\
So that men trembled, saying:  He forsooth\\
Has gone, has gone, and shall return no more.\\-
\end{poem}
```

> 1 I saw you starting for another war,
> 2   The emblem of adventure and of youth,
> 3   So that men trembled, saying: He for-
>   sooth
> 4 Has gone, has gone, and shall return no
>   more.

If you give a line number larger than the number of lines in a verse, it will be quietly ignored.

Note that `\poemvsindentlines` and `poemindentevery` will *stack*; if a line qualifies for both, it will be indented twice.

`\poemvsindentlines` allows you to arrange the indentation on each verse with very fine detail, assuming that all your verses are indented the same way.

`\poemverseskip`   Finally, you should meet `\poemverseskip`, which controls the distance between verses (stanzas) of the poem. By default, this is set equal to `\baselineskip`, leaving about one full line between stanzas. This is a standard LaTeX length, and can be reset in the usual way.

2.3 Customizing Line Numbering

2.3.1 Basics of Line Numbering

Line numbers, unlike in `verse`, *are* typeset by default. To turn them off, just `\poemlinenumsfalse`   say `\poemlinenumsfalse`. Of course, issuing `\poemlinenumstrue` will do the `\poemlinenumstrue`   opposite.

```
\poemlinenumsfalse
\begin{poem}
O captain! my captain!  The fearful trip is done; \\
\poemlinenumstrue
```

```
The ship has weather'd every rack, the prize we sought is won;\\
The port is near, the bells I hear, the people all exulting, \\
While follow eyes the steady keel, the vessel grim and daring; \\-
\end{poem}
```

Which gives us:

O captain! my captain! The fearful trip is done;
The ship has weather'd every rack, the prize we sought is won;
3   The port is near, the bells I hear, the people all exulting,
4   While follow eyes the steady keel, the vessel grim and daring;

Notice that `poetry` actually sets the line number *before* the rest of the line; so when turning line numbers on again, you need to put \poemlinenumstrue one line earlier than you might expect.

Line numbers aren't just on or off, however; you can have them every certain
\poemlinenumsevery   number of lines, for example. The command for this is \poemlinenumsevery; note that this is a *counter*, so redefine it with \setcounter and friends. By default, its value is 1. The value works as one would expect; 2 means a number every two lines, 3 every three lines, and so on. For example:

```
\setcounter{poemlinenumsevery}{2}
\poemlinenumstrue
\begin{poem}
O captain! my captain!  The fearful trip is done; \\
The ship has weather'd every rack, the prize we sought is won;\\
The port is near, the bells I hear, the people all exulting, \\
While follow eyes the steady keel, the vessel grim and daring; \\-
\end{poem}
```

Gives us this:

O captain! my captain! The fearful trip is done;
2   The ship has weather'd every rack, the prize we sought is won;
The port is near, the bells I hear, the people all exulting,
4   While follow eyes the steady keel, the vessel grim and daring;

This can, of course, be set to any value, though typically 1–4 are reasonable ones. A sufficiently high number will essentially turn off line numbering; e.g., 84, as by the time you've seen the 84th line you won't remember what the last line number was. But I'm certainly not going to make it impossible.

Line numbers can be placed on either the right or the left. By default, they are
\poemlinenumrighttrue   placed on the left; to place them on the right, use \poemlinenumrighttrue; to put
\poemlinenumrightfalse   them back on the left, use the corresponding command \poemlinenumrightfalse.

```
\setcounter{poemlinenumsevery}{1}
```

```
\poemlinenumrighttrue
\begin{poem}
O captain! my captain!  The fearful trip is done; \\
\poemlinenumrightfalse
The ship has weather'd every rack, the prize we sought is won;\\
The port is near, the bells I hear, the people all exulting, \\
While follow eyes the steady keel, the vessel grim and daring; \\-
\end{poem}
```

Which gives us this (notice that right-side line numbers are placed after \poemmaxlinewd):

O captain! my captain! The fearful trip is done;   1

The ship has weather'd every rack, the prize we sought is won;   2

3   The port is near, the bells I hear, the people all exulting,

4   While follow eyes the steady keel, the vessel grim and daring;

Notice again that the line's line number is printed *first*, before the line; so you need to put this command one line earlier than you'd expect.

Please notice also that *the line numbers are* not *part of the line when calculating widths*. They are boxes which overhang the end or beginning of the line. Please keep that in mind when you're planning your formats.

2.3.2   Styling Line Numbers

As mentioned before, the line numbers are set in boxes which overhang either the left or right margins of the line, depending on the value of \ifpoemlinenumright (that is, on whether you've isseud \poemlinenumrighttrue or not). The size of that box, and how much it overhangs the margin, are both within your control.

By default, the box is 2em wide, and it's separated from the margin by 0pt. To change the space between the box and the line itself, use the standard LaTeX length \poemlinenumboxgap; below, see some lines from *Henry V* to demonstrate. The first two lines are set with default line number spacing; the second two with \poemlinenumboxgap set to 1em:

\poemlinenumboxgap

1   We few, we happy few; we band of brothers;

2   for he today who sheds his blood with me

3   shall be my brother; be he ne'er so vile

4   this day shall gentle his condition.

\poemlinenumboxwd   The width of the line number box itself is controlled by \poemlinenumboxwd, another standard LaTeX length which can be reset, as usual, with \setlength. Below, we have some more of *Henry V*, this time with varying line number box widths; the first two lines have the default, the second two have a width of 1em:

And gentlemen in England now abed   1

shall think themselves accurs'd they were not here,   2

<div align="center">

and hold their manhoods cheap whilst any speaks   3
that fought with us, upon St. Crispin's day.   4

</div>

\thepoemline   The line number itself is held in the LaTeX macro `\thepoemline`; so you can style this as you wish. However, in the event you want line numbers to look a certain way in the poem itself, but a different way when you use them elsehwere, \poemlinenumstyle   you can use `\poemlinenumstyle` to change their appearance. For example, the following uses `\def\poemlinenumstyle{\footnotesize\color{green}}`:

1   Thou see'st the world, Volumnius, how it goes;
2   our enemies have beat us to the pit.
3   It would be better to jump in ourselves,
4   than tarry till they push us.

By default, `\poemlinenumstyle` does nothing.

Labels can be placed on any line of poetry, and refer back to that line. Hyperlinks will work properly, if you're using `hyperref` to produce them. For example, "line 4" refers to line 4 of the quotation from *Hamlet* we gave you earlier. Make such references with the usual LaTeX `\ref` and `\label` commands.

3   Document Structure

`poetry` actually provides some significant structural commands, as well, which may make it easier to typeset larger amounts of poetry. It was inspired here more by `poetrytex` than by `verse`; indeed, this area is really where `poetrytex` shines. However, `poetry` does add some functionality to it. We'll explore first counters, then sectioning.

3.1   Other Counters

`poetry` provides several other counters, which will probably rarely be useful when typesetting poetry, but which are provided just in case.
verseline   Besides `\thepoemline`, we also have `verseline`, a counter which keeps track of what line in a given verse is currently being typeset. It's reset every time you call `\\!`, to signify the end of a verse (or, more properly for poetry, stanza). It's typeset with `\theverseline`, which can be redefined in the normal LaTeX ways. For example, to number each verse with the poem line number, a colon, and a verse line number, try:

`\renewcommand{\thepoemline}{\arabic{poemline}:\arabic{verseline}}`

You'll get something like this:

1:1   Whose woods these are I think I know.
2:2   His house is in the village though;
3:3   He will not see me stopping here

<div align="center">

4:4 To watch his woods fill up with snow.

5:1 My little horse must think it queer
6:2 To stop without a farmhouse near
7:3 Between the woods and frozen lake
8:4 The darkest evening of the year.

</div>

We also have a counter for the number of poems themselves; two, actually. First
`abspoemno` is `abspoemno`, which keeps track of the poem numbers throughout the entire document. `abspoemno` is typeset, as always with LaTeX counters, with `\theabspoemno`. This number can be useful with sets of poems, such as Shakespeare's sonnets; or with anthologies which include untitled poems, which might otherwise be difficult to list in a table of contents.

`poemno` Second, there is the *poem group* counter, or `poemno`. `poemno` equals `abspoemno` unless you're utilizing poem groups (for which see Section 3.4). Starting a new poem group resets `poemno` but not `abspoemno`.

We will see some poetry-specific sectioning commands in Section 3.3.2. These commands increment `poemno` and `abspoemno`; so if you're not using those sectioning commands, these numbers won't be incremented. If for some reason you don't use a visible title for your poem, or wish to use your own sectioning commands, but still want it counted for the purposes of these counters, you need to tell `poetry`
`\poemnew` so. Issue `\poemnew` prior to starting your `poem` environment; it will increment your counters without putting anything in your text.

These counters are formatted, as always, with the LaTeX `\the` commands: `\thepoemno`, `\theabspoemno`.

3.2 Other Variables

`poetry` gives a few variables that we can use to fill in gaps when we have them.
`\poemblanktitle`   `\poemblanktitle` is the word that `poetry` will use when it's asked to typeset a title but it doesn't have one; by default, this is set as `Untitled`. Reset this with `\def` or `\renewcommand`; e.g., `\renewcommand{\poemblanktitle}{Not Titled}`. Then, when you are expected to input a title and you don't have one (possible in, for example, the `\titlepoem` command, for which see Section 3.3), instead of printing "Untitled", `poetry` will print "Not Titled".
`\poemblankauthor`   There is a corresponding `\poemblankauthor`, which does the same thing. Its default value is `Anonymous`.

3.3 Metadata

All poems have authors, even if we don't know who; and some also have titles. In other words, poems have *metadata*, data about themselves, and `poetry` provides easy ways to deal with these things, whether we have the information or not. Sometimes it will be convenient to use the variables from Section 3.2 in these commands.

poetry keeps the issue of *having* the metadata separate from *typesetting* it. You can set these variables but never have them appear in the final document. In theory, this information could instead by embedded in the `pdf` for document readers to use, or things of that nature. At present, pdf doesn't have any means to embed metadata about individual parts of a document; however, should such a capability be added, poetry will be ready.

3.3.1   Setting Metadata Values

`\thepoemtitle` `\thepoemtitle` holds the currently defined poem title, if any. You might find this handy for running headers, TOC entries, and the like.

`\thepoemauthor` `\thepoemauthor` corresponds to `\thepoemtitle`, and is useful for the same sorts of things.

Most written works these days have *licenses*; poetry makes it easy to spec-
`\thepoemlicense` ify that, as well. The variable is `\thepoemlicense`, and defaults to the string `Public Domain`. To alter this, simply redefine it with `\def` or `\renewcommand`; e.g., `\renewcommand{\thepoemlicense}{CC-BY-SA}`. To change the default, re-
`\poemdefaultlicense` define `\poemdefaultlicense`.

Publication date can also be important, particularly when combined with cer-
`\thepoempubdate` tain types of licensing. To specify this, redefine `\thepoempubdate`, which defaults to the value of `\today`.

3.3.2   Using Metadata

The most obvious way of using the metadata you've set for a poem is to *typeset* it. poetry offers a number of ways to do this.

If you're only interested in printing the poem's title, you can use the
`\titlepoem` `\titlepoem` command:

$$\texttt{\textbackslash titlepoem } \{\langle \textit{title} \rangle\}$$

E.g., with the default settings:

`\titlepoem{\poemblanktitle}` | # Untitled

If you have multiple untitled poems to typeset, it may be convenient to use `\thepoemno` or `\theabspoemno` to different them; e.g.,

$$\texttt{\textbackslash titlepoem\{Untitled No.\textbackslash\ \textbackslash thepoemno\}}$$

`\titlepoem` will reset the variable `\thepoemtitle` to whatever argument you give it. This will be available later for other uses.

If you want to typeset both the title and the author, poetry provides you with
`\titleauthorpoem` `\titleauthorpoem`:

10

$$\texttt{\textbackslash titlepoem} \ \{\langle title\rangle\} \ \{\langle author\rangle\}$$

`\titleauthorpoem`
  `{\poemblanktitle}`
  `{\poemblankauthor}`
	Untitled
	Anonymous

    `\titleauthorpoem` resets both `\thepoemtitle` and `\thepoemauthor` to the values you submit.

    `poetry` also gives you hooks to control the appearance of these macros.
`\poemtitleprehook` `\poemtitleprehook` is inserted by `poetry` prior to the title text, and `\poemtitleposthook`
`\poemtitleposthook` is inserted afterward. For `\titleauthorpoem`, there is also `\poemauthorprehook`
`\poemauthorprehook` and `\poemauthorposthook`, inserted before and after the author. By default,
`\poemauthorposthook` the `pre` commands simply `\begin{center}`, while the `post` commands contain
`\end{center}`, but you can certainly change their values to achieve the appearance you want:

<div align="right">

Untitled   Anonymous

</div>

```
 \def\poemtitleprehook{%
\bigskip\noindent}
 \def\poemtitleposthook{}
 \def\poemauthorprehook{\hfill}
 \def\poemauthorposthook{\bigskip}
 \titleauthorpoem{\poemblanktitle}{\poemblankauthor}
```

    These settings are rather naïve, since they don't take the length of either author or title into account, but they suffice to show the power of using these hooks.
`\poemtitlestyle`   There is also `\poemtitlestyle`, which sets the styling of the title. By default, it is simply set with `\def\poemtitlestyle{\LARGE\bfseries}`.
`\poemauthorstyle`   Similarly, there is `\poemauthorstyle`, which provides the styles for the author (if one will be typeset). By default, this is `\def\poemauthorstyle{\Large}`.

3.4   Structural Features

In addition to typesetting poems individually, `poetry` has some features to facilitate typesetting collections of poetry. Structurally, primarily we're dealing with poem groups (for which see Section 3.4.1); however, we also have significant power in indexing and tracking poems.

3.4.1 Poem Groups

Poem groups are just what they sound like: groups of poems. They might be arranged by topic, by author, by style, or by any other criteria. As we have already seen in Section 3.1, the `poemno` counter tracks the number of poems individually within poem groups. Let's see a few of the commands we have for creating and dealing with these groups.

`\poemgroup` `\poemgroup` starts a new poem group; it resets the `poemno` counter and adds one to the poem group counter. That's it; it's a purely structural command. Formatting title pages and the like for poem groups is another matter. `\endpoemgroup` `\endpoemgroup` can optionally be used to end the poem group. Like `\poem`, however, these can also be used like a LATEX environment, using `\begin{poemgroup}` and `\end{poemgroup}`.

`poemgroupno` `poemgroupno` is the counter which keeps track of which poem group you're currently in. It is a standard LATEX counter, which can be reset, manipulated, formatted, and displayed in the usual ways.

`\poemgroupheading` `\poemgroupheading` prints a heading for the poem group. This may or may not be on its own page; that's up to your formatting.

$$\texttt{\textbackslash poemgroupheading }\{\langle\textit{title}\rangle\}\ \{\langle\textit{other}\rangle\}$$

Both these arguments are mandatory, though of course one or the other might be empty. The first is the title of the poem group; the second is whatever else. It may be the author of the group; the style of poetry (e.g., anapestic, or alliterative); or any other criteria by which the poetry is broken into groups.

`\thepoemgroupname` `\poemgroupheading` also resets the value of `\thepoemgroupname`, which is another bit of metadata that can used throughout that group. It resets it to the value of its first argument. If it is not reset, its value is simply the value `\poemgroupblankname` of `\poemgroupblankname`, which defaults to `Untitled Group`. By redefining this command, you can change the default group name. You can also redefine `\thepoemgroupname` separately, but be aware that calling `\poemgroupheading` will reset it again.

`\poemgrouptitleprehook` The hooks listed to the left are inserted where their names imply into the `\poemgrouptitleposthook` heading. By default, all are defined to do nothing, but you can fix this with `\def` `\poemgroupotherprehook` or `\renewcommand`. A common styling would involve putting `\begin{center}` `\poemgroupotherposthook` and `\end{center}` into one or both of the `pre` and `post` hooks, respectively. Another common requirement would involve putting `\cleardoublepage` into `\poemgrouptitleprehook`, along with whatever else is required, in order to ensure a poem group starts on the next new recto (right-side and odd-numbered) page.

`\poemgrouptitlestyle` The title can be styled by redefining `\poemgrouptitlestyle`, which by default simply contains `\Huge`. The "other" argument can be styled with `\poemgroupotherstyle` `\poemgroupotherstyle`, which by default contains simply `\LARGE`.

3.4.2 List of Poems

To go along with the table of contents, the list of figures, and the list of tables, `poetry` provides a list of poems. We use standard LATEX facilities to produce

this list, so they should be pretty familiar to you. The source below can give the nitty-gritty.

\listofpoems   Just add `\listofpoems` wherever you want your list of poems to appear. Unlike `\tableofcontents` and the like, `\listofpoems` does *not* start a new chapter or section, change the headings, and so forth; since `poetry` doesn't know what document class you're using, or how you want the list set up, it leaves the details up to you. This is easy enough to elaborate on; for example, to set up the list of poems like an unmarked chapter, like the table of contents:

```
\chapter*{\lopname}
\listofpoems
```

\lopname   `\lopname` is a macro which simply contains whatever you want the title of the list of poems to be; by default, it is `List of Poems`.

   `poetry` also provides a little syntactic sugar to make adding entries to the list of poems easier. Rather than doing `\addcontentsline` and the like, simply use \addtolop `\addtolop`:

$$\text{\texttt{\textbackslash addtolop} } \{\langle \textit{contents-line}\rangle\}$$

   This will add an entry to the list of poems in the default style; this is set up much like a normal section entry without indentation. To alter this, redefine \poemlopformat `\poemlopformat`. By default, this is defined in a very LaTeX-y way; the standard definition is shown on top below, with a different one underneath it:

```
\def\poemlopformat{\@dottedtocline{1}{0em}{2.3em}}
\def\poemlopformat#1#2{\hbox to\linewidth{\textit{#1}\hfill#2}}
```

   You can specify it however you like.

\poemgrouplopformat   There is an equivalent `\poemgrouplopformat`, which by default is designed to mimic `\chapter` lines in the standard LaTeX classes. This can be redefined as \addgrouptolop necessary, too. You add a group to the list of poems by using `\addgrouptolop`, which functions just like `\addtolop` does. So something like this:

```
\def\poemgrouplopformat#1#2{\hbox to\linewidth{#1\hfil#2}}
\addgrouptolop{Poem Group Name}
```

   Note that none of our sectioning commands insert themselves into the table of contents or the list of poems by default. If you want this to happen automatically, best to write small wrappers macros around them. E.g.:

```
\def\makenewpoem#1{%
\titlepoem{#1}%
\addtolop{#1}%
}%
```

3.4.3 Index of First Lines

Few people remember "Sonnet 18," but many people remember, "Shall I compare thee to a summer's day?" For this reason, an index of first lines is an important part of most books of poetry. `poetry` provides facilities to easily create them.

poetry uses the `imakeidx` package to produce this index, so you can use the options it contains. To run the index, `poetry` uses the following configuration line:

```
\makeindex[intoc,columns=1,name={\jobname-iofl},title={\poemioflname}]
```

Obviously, you can change this if you need to. This means that, by default, the index will appear in the table of contents (if there is one); the index will be set in one column; the index file name will be `\jobname-iofl` (`\jobname` is a TeX variable that contains the name of your document, without its extension), and the title of the index (what appears on the top of the page) will be the value of the command `\poemioflname`.

\poemioflname   To change the title that appears at the top of the index, change `\poemioflname` to whatever you want; e.g., `\def\poemioflname{Index}`.

\printiofl   Use `\printiofl` at whatever point in your document you want the index to actually appear.

\poemfirstline   To make entries in the index, you have two options. First, use `\poemfirstline`, which takes a single argument, which is the entry you want to appear in the index. This command will *not* typeset its argument in the document, only send it to the

\iofl   index. Second, you can use `\iofl`, which also takes a single argument; however, it will not only enter that line into the index, but it will also typeset it in the document. This means that you can wrap the first line of your poem in `\iofl` (*without* the \\), and it will put it both in your poem and your index, without you having to type it twice.

4   Implementation

```
1 \ProvidesPackage{poetry}%
2 \newcounter{poemline}%
3 \newcount\poemlineno%
4 \newcounter{verseline}%
5 \newcount\vslineno%
6 \newcounter{abspoemno}%
7 \newcount\poem@abspoemno%
8 \newcounter{poemno}%
9 \newcount\poem@poemno%
10 \newcounter{poemgroupno}\setcounter{poemgroupno}{0}%
11 \newcount\poem@groupno\poem@groupno=0%
12 \newcounter{poemlinenumsevery}\setcounter{poemlinenumsevery}{1}%
13 \newcount\poem@linenumsevery\poem@linenumsevery=\value{poemlinenumsevery}%
14 \newlength{\poemindent}\setlength{\poemindent}{0pt}%
15 \newlength{\poemtopskip}{\setlength{\poemtopskip}{1em}%
```

```
16 \newlength{\poembotskip}{\setlength{\poembotskip}{1em}%
17 \newdimen\poem@topskip\newdimen\poem@botskip%
18 \newdimen\poem@defleftskip\poem@defleftskip=\the\poemindent%
19 \newdimen\poem@maxlinewd%
20 \newlength{\poemmaxlinewd}\setlength{\poemmaxlinewd}{\textwidth}%
21 \newdimen\poem@hinwd\poem@hinwd=1em%
22 \newlength{\poemhinwd}\setlength{\poemhinwd}{1em}%
23 \newif\ifpoemlinenums\poemlinenumstrue%
24 \newif\ifpoemlinenumright\poemlinenumrightfalse%
25 \newdimen\poem@linenumboxwd\poem@linenumboxwd=2em%
26 \newlength{\poemlinenumboxwd}\setlength{\poemlinenumboxwd}{2em}%
27 \newdimen\poem@linenumboxgap\poem@linenumboxgap=0pt%
28 \newlength{\poemlinenumboxgap}\setlength{\poemlinenumboxgap}{0pt}%
29 \def\poemlinenumstyle{}%
30 \newdimen\poem@hangindent\poem@hangindent=1em%
31 \newlength{\poemhangindent}\setlength{\poemhangindent}{1em}%
32 \newcount\poem@indentevery\poem@indentevery=0%
33 \newcounter{poemindentevery}\setcounter{poemindentevery}{0}%
34 \newdimen\poem@verseskip\poem@verseskip=\baselineskip%
35 \newlength{\poemverseskip}\setlength{\poemverseskip}{\baselineskip}%
36 \newdimen\poem@buff%
37 \newcount\poem@tmp%
38 % from David Carlisle's answer on tex.stackexchange.com,
39 % question 57598; allows us to loop through a string in a
40 % command.  We will use this for the verse indentation
41 % counter, as well as for the generic line indentatin
42 % counter, both in |\placelineno|.
43 \def\poem@expandvsloop#1{\poem@xvsloop#1\relax}%
44 \def\poem@xvsloop#1{%
45 \ifx\relax#1\else%
46 \poem@tmp=#1%
47 \ifnum\poem@tmp=\the\vslineno%
48 \hin%
49 \fi%
50 \expandafter\poem@xvsloop%
51 \fi%
52 }%
53 % from verse.sty
54 \newcommand{\poem@ifbang}[1]{\@ifnextchar !{\@firstoftwo{#1}}}
55 \newcommand{\poem@ifhyph}[1]{\@ifnextchar -{\@firstoftwo{#1}}}
56 \newcommand{\poem@ifstar}[1]{\@ifnextchar *{\@firstoftwo{#1}}}
57 \newcommand{\poem@ifques}[1]{\@ifnextchar ?{\@firstoftwo{#1}}}
58 % get modulus
59 \newcount\poem@tmpa
60 \def\modulo#1#2{%
61 \poem@tmpa=#1%
62 \divide\poem@tmpa by#2%
63 \multiply\poem@tmpa by#2%
64 \multiply\poem@tmpa by-1%
65 \advance\poem@tmpa by#1\relax%
```

```
66 }%
67 \def\poemvsindentlines{}%
68 \def\poemlnindentlines{}%
69 \def\tlap#1{\vbox to0pt{\vss\hbox{#1}}}%
70 \def\blap#1{\vbox to0pt{\hbox{#1}\vss}}%
71 \def\placelineno{%
72 \setcounter{verseline}{\the\vslineno}%
73 \setcounter{poemline}{\the\poemlineno}%
74 \poem@linenumsevery=\value{poemlinenumsevery}%
75 \poem@linenumboxgap=\the\poemlinenumboxgap%
76 \poem@linenumboxwd=\the\poemlinenumboxwd%
77 \modulo{\the\poemlineno}{\the\poem@linenumsevery}%
78 \ifpoemlinenums%
79 \ifnum\poem@tmpa=0%
80 \ifpoemlinenumright%
81 \hskip0pt\tlap{%
82 \rlap{%
83 \hskip\poem@maxlinewd%
84 \hskip\poem@linenumboxgap%
85 \hbox to\poem@linenumboxwd{%
86 \hfil%
87 \poemlinenumstyle\thepoemline%
88 }%
89 }%
90 }%
91 \else%
92 \hskip-\poem@linenumboxgap%
93 \llap{%
94 \tlap{%
95 \hbox to\poem@linenumboxwd{%
96 \poemlinenumstyle\thepoemline%
97 \hfil%
98 }\penalty10000%
99 }%
100 }\penalty10000%
101 \fi%
102 \else
103 \hskip-\poem@linenumboxgap%
104 \llap{\tlap{\hbox to\poem@linenumboxwd{\hfil}}}%
105 \penalty10000%
106 \fi%
107 \else
108 \hskip-\poem@linenumboxgap%
109 \llap{\tlap{\hbox to\poem@linenumboxwd{\hfil}}}%
110 \penalty10000%
111 \fi%
112 \par\vskip-\baselineskip%
113 \poem@indentevery=\value{poemindentevery}%
114 \ifnum\poem@indentevery=0%
115 \else%
```

```
116 \modulo{\the\poemlineno}{\the\poem@indentevery}%
117 \ifnum\poem@tmpa=0%
118 \hin%
119 \fi%
120 \fi%
121 \expandafter\poem@expandvsloop\expandafter{\poemvsindentlines}%
122 \def\@currentlabel{\thepoemline}%
123 \phantomsection%
124 }%
125 \def\poem@endline{%
126 \par%
127 \advance\poemlineno by1%
128 \advance\vslineno by1%
129 \poem@defaultpars%
130 \leftskip=\poem@defleftskip%
131 \placelineno%
132 }%
133 \def\poem@endpart{%
134 \par%
135 \advance\poemlineno by1%
136 \advance\vslineno by1%
137 \poem@defaultpars%
138 \leftskip=\poem@defleftskip%
139 }%
140 \def\poem@resume{%
141 \placelineno%
142 }%
143 \def\*{%
144 \poem@resume%
145 }%
146 \def\poem@endverse{%
147 \par%
148 \vslineno=1%
149 \advance\poemlineno by1%
150 \poem@verseskip=\the\poemverseskip%
151 \vskip\poem@verseskip%
152 \leftskip=\poem@defleftskip%
153 \placelineno%
154 }
155 \def\poem@endversepart{%
156 \par%
157 \advance\poemlineno by1%
158 \vslineno=1%
159 \poem@defaultpars%
160 \leftskip=\poem@defleftskip%
161 }
162 \def\poem@defaultpars{%
163 \poem@hangindent=\the\poemhangindent%
164 \everypar={%
165 \parindent=0pt%
```

```
166 \hangindent=\poem@hangindent%
167 \hangafter=1%
168 }%
169 }%
170 \def\hin{%
171 \poem@hinwd=\the\poemhinwd%
172 \advance\leftskip by\the\poem@hinwd%
173 \penalty10000%
174 }%
175 \def\poemnew{%
176 \advance\poem@poemno by1\stepcounter{poemno}%
177 \advance\poem@abspoemno by1\stepcounter{abspoemno}%
178 }%
179 \def\poem{%
180 \widowpenalty=0\clubpenalty=0%
181 \def\\{%
182 \poem@ifbang{\poem@endverse}{%
183 \poem@ifhyph{\par\vskip\poem@botskip\relax}{%
184 \poem@ifstar{\poem@endpart}{%
185 \poem@ifques{\poem@endversepart}{\poem@endline}%
186 }%
187 }%
188 }%
189 }%
190 \poem@defleftskip=\the\poemindent%
191 \leftskip=\poem@defleftskip%
192 \poem@maxlinewd=\the\poemmaxlinewd%
193 \hsize=\the\poem@defleftskip%
194 \advance\hsize by\the\poem@maxlinewd%
195 \poemlineno=1%
196 \vslineno=1%
197 \poem@topskip=\the\poemtopskip%
198 \poem@botskip=\the\poembotskip%
199 \vskip\poem@topskip%
200 \poem@defaultpars%
201 \noindent%
202 \poem@ifstar{}{\placelineno}%
203 }%
204 \def\endpoem{%
205 }%
206 \newdimen\poem@centlinewd
207 \def\centerpoemon#1{%
208 \settowidth{\poem@centlinewd}{#1}%
209 \ifdim\poem@centlinewd>\linewidth\poem@defleftskip=0pt%
210 \else
211 \poem@buff=\linewidth%
212 \advance\poem@buff by-\poem@centlinewd%
213 \divide\poem@buff by2%
214 \poem@defleftskip=\the\poem@buff%
215 \poemindent=\the\poem@buff%
```

```
216 \poem@maxlinewd=\the\poem@centlinewd%
217 \setlength{\poemmaxlinewd}{\poem@maxlinewd}%
218 \fi
219 }%
220 \def\centerpoemoff{%
221 \poem@defleftskip=0pt\setlength{\poemindent}{0pt}%
222 \poem@maxlinewd=\the\linewidth%
223 \setlength{\poemmaxlinewd}{\linewidth}%
224 }%
```

Define a few variables for document metadata.

```
225 \def\thepoemtitle{\poemblanktitle}%
226 \def\thepoemauthor{\poemblankauthor}%
227 \def\thepoemlicense{Public Domain}%
228 \def\poemdefaultlicense{Public Domain}%
229 \def\thepoempubdate{\today}%
```

Here we begin the structural code; which, of course, uses a lot of the code above, particularly the counters. First, let's give folks a way to title their poems. But even before that, we need to define some more commands to help people style them.

```
230 \def\poemblanktitle{Untitled}%
231 \def\poemblankauthor{Anonymous}%
232 \def\poemtitlestyle{\LARGE\bfseries}%
233 \def\poemauthorstyle{\Large}%
234 \def\poemtitleprehook{%
235 \begin{center}%
236 }%
237 \def\poemtitleposthook{%
238 \end{center}%
239 }%
240 \def\poemauthorprehook{%
241 \begin{center}%
242 }%
243 \def\poemauthorposthook{%
244 \end{center}%
245 }%
```

Specifically, one for titles with no authors:

```
246 \def\titlepoem#1{%
247 \advance\poem@poemno by1\stepcounter{poemno}%
248 \advance\poem@abspoemno by1\stepcounter{abspoemno}%
249 \def\thepoemtitle{#1}\def\thepoemauthor{\poemblankauthor}%
250 \begingroup%
251 \poemtitleprehook%
252 \poemtitlestyle #1%
253 \poemtitleposthook%
254 \endgroup%
255 }%
```

Next, one for titles and authors:

```
256 \def\titleauthorpoem#1#2{%
```

```
257 \advance\poem@poemno by1\stepcounter{poemno}%
258 \advance\poem@abspoemno by1\stepcounter{abspoemno}%
259 \def\thepoemtitle{#1}\def\thepoemauthor{#2}%
260 \begingroup%
261 \poemtitleprehook%
262 \poemtitlestyle #1%
263 \poemtitleposthook%
264 \endgroup%
265 \begingroup%
266 \poemauthorprehook%
267 \poemauthorstyle #2%
268 \poemauthorposthook%
269 \endgroup%
270 }%
```

Now we define the poem group commands. These are pretty simple, essentially just resetting various variables to sensible values. We have the `\endpoemgroup` command here just to allow people to use them as environments, if they want.

```
271 \def\poemgroup{%
272 \advance\poem@groupno by1%
273 \stepcounter{poemgroupno}%
274 \poem@poemno=0\setcounter{poemno}{0}%
275 \def\thepoemgroupname{\poemgroupblankname}%
276 }%
277 \def\endpoemgroup{%
278 }%
279 \def\poemgrouptitleprehook{}%
280 \def\poemgrouptitleposthook{}%
281 \def\poemgroupotherprehook{}%
282 \def\poemgroupotherposthook{}%
283 \def\poemgrouptitlestyle{\Huge}%
284 \def\poemgroupotherstyle{\LARGE}%
285 \def\poemgroupblankname{Untitled Group}%
286 \def\poemgroupname{\poemgroupblankname}%
287 \def\poemgroupheading#1#2{%
288 \def\thepoemgroupname{#1}%
289 \begingroup%
290 \poemgrouptitleprehook%
291 \poemgrouptitlestyle #1%
292 \poemgrouptitleposthook%
293 \endgroup%
294 \begingroup%
295 \poemgroupotherprehook%
296 \poemgroupotherstyle #2%
297 \poemgroupotherposthook%
298 \endgroup%
299 }%
```

Now let's define the necessary commands for the list of poems. (*Not* the index of first lines, yet.)

```
300 \def\poemlopformat{%
```

```
301 \@dottedtocline{1}{0em}{2.3em}%
302 }%
303 \def\l@poem{\poemlopformat}%
304 \def\addtolop#1{%
305 \addcontentsline{lop}{poem}{#1}%
306 }%
307 \def\addgrouptolop#1{%
308 \addcontentsline{lop}{poemgroup}{#1}%
309 }%
310 \def\lopname{List of Poems}%
311 \def\listofpoems{%
312 \@starttoc{lop}%
313 }%
314 \def\poemgrouplopformat#1#2{%
315 \ifnum \c@tocdepth >\m@ne
316 \addpenalty{-\@highpenalty}%
317 \vskip 1.0em \@plus\p@
318 \setlength\@tempdima{1.5em}%
319 \begingroup
320 \parindent \z@ \rightskip \@pnumwidth
321 \parfillskip -\@pnumwidth
322 \leavevmode \bfseries
323 \advance\leftskip\@tempdima
324 \hskip -\leftskip
325 #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
326 \penalty\@highpenalty
327 \endgroup
328 \fi%
329 }%
330 \def\l@poemgroup{\poemgrouplopformat}%
```

Now let's set up the index of first lines. We use the `imakeidx` package for this, so as not to interfere with any existing indices.

```
331 \RequirePackage{imakeidx}
332 \makeindex[intoc,columns=1,name={\jobname-iofl},title={\poemioflname}]
333 \def\poemioflname{Index of First Lines}%
334 \def\poemfirstline#1{%
335 \index[\jobname-iofl]{#1}%
336 }%
337 \def\iofl#1{%
338 \index[\jobname-iofl]{#1}%
339 #1%
340 }%
341 \def\printiofl{%
342 \printindex[\jobname-iofl]%
343 }%
```

Index