

---

## An attempt at ragged-right typesetting

Udo Wermuth

### Abstract

Plain  $\TeX$  (as well as  $\LaTeX$ ) provides the command `\raggedright` to typeset text *unjustified*, that is, only the left margin is straight, the right margin varies in its length and looks ragged. The command gives good results for narrow columns, for example, when it is applied in marginal notes and short quotations. However, its output addresses only the English tradition. For wider measures it produces the ragged-right output that is called “Rauhsatz” in German and not the more demanding “Flattersatz”. In this article we examine what recommendations are formulated by a German typographer to typeset text in “Flattersatz” and then make an attempt to capture these recommendations in  $\TeX$  macros.

### 1 Introduction

By default, plain  $\TeX$  produces paragraphs that are justified. In most cases it is able to produce decent output if the measure allows that at least 60 characters, punctuation marks, and blank spaces fit into one line. With some care the length of the line might be reduced to only 40 symbols and blank spaces and acceptable output can still be created. Smaller line widths should use *unjustified text* [3, p. 27]. (Most of the time this article uses in its experiments a measure of 150 pt which allows an average of  $\approx 32$  characters per line in `cmr10` whose lowercase alphabet length is 127.5837 pt; see [3, p. 28].)

For languages that are written from left-to-right *unjustified* text usually means that the left-hand side has a straight margin as for justified text and the right-hand margin is ragged as line breaks occur at the end of the line when the line is sufficiently filled. There are several phrases to describe this situation: *flush-left typesetting*, *ragged-right typesetting*, *unjustified*; all have the same meaning.

In the world of  $\TeX$ , unjustified is often called ragged-right typesetting as plain  $\TeX$  defines the control word `\raggedright` to typeset paragraphs unjustified [8, p. 356].  $\LaTeX$  provides a *flushleft* environment and a `\raggedright` declaration; see [11, pp. 111–112]. The latter control word has a different implementation than the plain  $\TeX$  version; the next section will discuss this further.

For centuries, book printing has applied justification to nearly all paragraphs with one important exception: epics or poetry in general. These texts

were often printed unjustified. But in the modern interpretation of traditional typesetting, book designers and typographers started to apply ragged-right margins to all kinds of books, occasionally even novels, at least in the 20th century [16, p. 20]. As in the case of justification, certain best practices and guidelines to avoid bad looking results have been formulated to typeset such texts unjustified.

**Motivation.** I am aware that ragged-right typesetting is a difficult topic. Once I needed ragged-right typesetting for a German text and the  $\TeX$  macro `\raggedright` didn’t give a result that I liked (too many hyphens) and I fixed it manually. Much later I looked up unjustified typesetting in my books about typography, and tried to develop macros that implement ragged right for a larger variety of measures. I describe some of them in this article, but I have to admit that my experience with them is still limited. And note, this is the first time that I apply them to an English text. Thus, I look at this article as a start to the discussion of how to approach the problem of automated ragged-right typesetting with  $\TeX$ .

This paper discusses neither the readability of unjustified margins nor the situations when they shall be applied. A lot of factors are involved but here the assumption is made that the designer has determined that the needs of the text request or at least allow ragged-right typesetting.

**Two words in German.** German typographers translate “ragged-right margins” with two German words: *Rauhsatz* and *Flattersatz*. (“Satz” is typesetting; “rauh” has here the meaning “raw” or “untreated”; “flattern” means “to flutter”.) The two German words are not synonyms but rather show two different sets of rules—or better, *recommendations*—formulated to distinguish layouts by the significance of the visual effect of the right margin. Their difference is very important as the former is considered low quality typesetting compared to the latter. (“Rauhsatz damages the language, it is only acceptable for cheap, short-lived works” [16, p. 90]; my translation.) *Rauhsatz* is used with small line widths; *Flattersatz* is applied for wider measures or as an alternative for justified text. It requests that the right margin forms a rhythmical pattern: Lines shall alternate in length. Precise descriptions follow in section 4.

I don’t know why English typographers use only one word instead of the two that are considered necessary by German typographers. It might be that the concept of “Flattersatz” is so distinct from the English ragged-right typesetting as the line lengths vary so much—probably due to the longer average

word length in German — that no separate English word is needed because the effect doesn't independently stand out in English typesetting. But as the traditions for different languages have different recommendations it might be useful to have a more complicated ragged-right typesetting macro that is flexible enough to reflect these diverse traditions.

**Contents.** First, section 2 discusses some available implementations in T<sub>E</sub>X to typeset text unjustified and section 3 presents a few examples of bad ragged-right output. Section 4 describes some recommendations, and section 5 tries to capture them in T<sub>E</sub>X macros. Modern T<sub>E</sub>X installations run fast and we can throw more code at the problem than in earlier days. A general approach to define a ragged-right typesetting macro is introduced in section 6. Section 7 applies the results to English and German examples. Control sequences to set line breaks manually in this context are the topic of section 8. A short summary in section 9 concludes the article.

## 2 What do T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X implement?

First, the definition for ragged-right typesetting in plain T<sub>E</sub>X is discussed. Actually, there are two definitions in `plain.tex` [8, pp. 355–356]. I change the format and replace a constant-value register by its content to enhance readability. Plain T<sub>E</sub>X does not have a command that switches back to justification so that the command is normally used in a group, else it is active for the rest of the document.

```
\def\raggedright{\rightskip=0pt plus 2em
  \spaceskip=0.3333em \xspaceskip=0.5em\relax}
\def\ttraggedright{\tt
  \rightskip=0pt plus 2em\relax}
```

How does the first definition work? The line-breaking algorithm of T<sub>E</sub>X includes two globs of glue, the `\leftskip` and the `\rightskip`, which are placed at the beginning and the end of every created line [8, p. 100]. The default value in plain T<sub>E</sub>X for both skips is 0pt, so they have no effect. Here the `\rightskip` keeps the natural width of 0pt but adds stretchability. This allows a line to end before it covers the entire width of `\hsize`.

The next two assignments change how T<sub>E</sub>X determines the amount of white space that it puts between words. Usually these *interword spaces* are based on the values of certain `\fontdimen` parameters defined by the current font. But the above assignments make interword spaces a fixed width, with a larger value after colons, periods, exclamation marks, and question marks. More precisely, the `\spaceskip` defines the interword space while the `\xspaceskip` sets the space after characters with a

space factor  $\geq 2000$  [8, p. 76 and p. 351]. The values are given in the unit “em” to set them with respect to the font that is active when the macro is called. Only this font should be used in the text. (If more fonts are involved, then all `\fontdimen3` and `\fontdimen4` values for interword stretch and shrink should be cleared and `\spaceskip` shouldn't be used.) For `cmr10`, `\spaceskip`'s natural width equals the value of the font's interword space, i.e., `\fontdimen2`, but the `\xspaceskip` is larger than the sum of `\fontdimen2` and `\fontdimen7`, the extra space, which is 0.44444em; see [8, p. 433].

It is a specialty of American English typesetting to have more white space after certain punctuation marks; in [3, pp. 29–30], this practice is classified deprecated as space itself is a kind of punctuation. The article [13] represents a longer example of ragged-right typesetting by an American typographer and type designer; it applies extra space after end-of-sentence periods. For German texts, the book [4, p. 183] requests the normal interword space after the period for the end of the sentence. But an old rule is mentioned that recommends setting a larger white space manually in certain circumstances to support the identification of a sentence's end: a small interword space and an abbreviation that ends with a period and that occurs at the end of a sentence.

The second definition does essentially the same for monospaced fonts as the first does for proportional fonts. The former has by default a fixed-width interword space, so the assignments to the parameters `\spaceskip` and `\xspaceskip` can be omitted. Note however that the font is explicitly selected before the assignment to `\rightskip` is made to get the correct value for the unit “em”.

On page 338 of *The METAFONTbook* [9] code from the file `testfont.tex` is shown that “improves on plain T<sub>E</sub>X's `\raggedright`”:

```
\ifdim\fontdimen6\testfont<10pt
  \rightskip=0pt plus 20pt
\else
  \rightskip=0pt plus 2em
\fi
\spaceskip=\fontdimen2\testfont
\xspaceskip=\fontdimen2\testfont
  \advance\xspaceskip by \fontdimen7\testfont
```

The improvements are: `\rightskip`'s stretchability is set to  $\min(20\text{pt}, 2\text{em})$  and interword spaces use the relevant `\fontdimen` parameters of the font that is tested.

L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub>'s `\raggedright` uses a similar setup. Again I format the code for the column width and replace constant-valued registers by their contents.

```
% see ltmiscen.dtx
\newskip\@rightskip
  \@rightskip=0pt plus 0pt minus 0pt
\def\raggedright{\let\=\@centercr
  \@rightskip=0pt plus 1fil
  \rightskip=\@rightskip
  \leftskip=0pt plus 0pt minus 0pt
  \parindent=0pt}
```

Two backslashes are used to implement a line break; see [11, p.112]. This aspect of the macro is not analyzed further in this article.

As in the plain  $\TeX$  macro, `\rightskip` gets a new value. Here it is done in two steps involving the skip register `\@rightskip`. The first difference is that infinite stretchability is assigned, which makes very short lines acceptable. It creates “very ragged-looking paragraphs” [12, p.103]. The second difference is that nothing is done to the interword spaces, they keep their stretch- and shrinkability. Moreover, the macro suppresses indentation.

### 3 What is bad ragged-right typesetting?

Of course, we are interested in good ragged-right typesetting. But the word “good” is sometimes difficult to define whereas it is quite easy to show some aspects that should not happen when a paragraph is typeset ragged right. So let’s learn first what should be avoided, at least according to German tradition.

To start, the right margin shouldn’t create a contour like an arrowhead, i.e., lines get longer in small steps and after the longest line the next lines decrease their length in small steps. Such structures are easily spotted by a reader and distract him from the text. (In the case that such a shape is wanted, use  $\TeX$ ’s `\parshape` [8, p.101] or a package like `shapepar` from CTAN.)

#### Example 1: Description

A short plain  $\TeX$  example of ragged-right typesetting with several criticizable aspects.

#### $\TeX$ input

```
\ninepoint\raggedright\noindent
The text of this example is typeset ragged
right with the macro of plain  $\TeX$ ; it is
designed to demonstrate some bad things that
happen if the line breaks are not manually
improved. For example, the first three lines
and the last three lines build a pair of
inverse slopes.
```

#### $\TeX$ output

The text of this example is typeset ragged right with the macro of plain  $\TeX$ ; it is designed to demonstrate some bad things that happen if the line breaks are not manually improved. For example, the first three lines and the last three lines build a pair of inverse slopes.  $\square$

The macro `\ninepoint` is defined in `tugboat.sty`.

In the next two paragraphs the line lengths vary and don’t build a contour; but other problems occur.

#### Example 1 continued: $\TeX$ input

```
\noindent
Another kind of problem is shown in the next
line: The last word sticks out as the other
two lines are a little bit shorter; it looks
similar to the first lines.
```

```
\noindent
```

Surprise! This is a new paragraph. But as it is not indented it is very hard to see that in the output.

#### $\TeX$ output

Another kind of problem is shown in the next line: The last word sticks out as the other two lines are a little bit shorter; it looks similar to the first lines. Surprise! This is a new paragraph. But as it is not indented it is very hard to see that in the output.  $\square$

Of course, the “a” at the end of one line and the identification of the last line of the paragraph should be improved. In [7], Fig. 5 [10, p.83], the first situation is called the “sticking-out problem”.

The problem that the start of a paragraph is not clearly signaled occurs with justified typesetting too. And in justified texts it can happen that one word is repeated at the start or the end of several consecutive lines. Such *stacks* can appear too if the text is typeset ragged right. But it can also happen that the text itself forms an unintended shape.

#### Example 1 continued: $\TeX$ input

```
\noindent
This text creates a known pattern at its right
end. A twist of fate\Dash or it happens to be bad
luck with \TUB’s measure. No, not at all, I
designed the text. I {\sl placed\} the
uppercase letters at the {\sl end\} of the
lines; I wanted to make them build this pattern.
```

#### $\TeX$ output

This text creates a known pattern at its right end. A twist of fate—or it happens to be bad luck with *TUGboat*’s measure. No, not at all, I designed the text. I *placed* the uppercase letters at the *end* of the lines; I wanted to make them build this pattern.  $\square$

Okay, we have seen a few problems which might occur with ragged-right typesetting but not with justified text and we have seen that some problems of justified typesetting can also appear with ragged right. The three problems of the first group that we saw are: Either the right margin or the words at the line ends build a shape and a single word might stick out into the margin. To the common problems belong the identification of the start of a paragraph if it has no indentation and the occurrence of stacks.

Of course the output is so bad that the whole text of the example must be reformatted. How to fix the situation? In the first paragraph a `\break` between “are not” avoids any contour. A tie after the colon fixes the second paragraph. A break between “not indented” in the third paragraph and one between “text. I” together with a tie in “I wanted” in the fourth improves the rest of the output.

**Example 1 continued: T<sub>E</sub>X output improved**

The text of this example is typeset ragged right with the macro of plain T<sub>E</sub>X; it is designed to demonstrate some bad things that happen if the line breaks are not manually improved. For example, the first three lines and the last three lines build a pair of inverse slopes.

Another kind of problem is shown in the next line: The last word sticks out as the other two lines are a little bit shorter; it looks similar to the first lines.

Surprise! This is a new paragraph. But as it is not indented it is very hard to see that in the output.

This text creates a known pattern at its right end. A twist of fate—or it happens to be bad luck with *TUGboat*’s measure. No, not at all, I designed the text.

I placed the uppercase letters at the *end* of the lines; I wanted to make them build this pattern. □

The output is not perfect yet but much better than before. This little example shows what typographers state about the manual rework of line breaks: The processing of ragged-right texts is laborious and the time expenditure is higher than with justified text [4, p. 152].

**A closer look at the sticking-out problem.**

We saw that plain T<sub>E</sub>X’s `\raggedright` sets fixed-width interword spaces and assigns stretchability to `\rightskip`. With this setup the sticking-out problem cannot occur very often.

First, let’s define what the phrase “sticking-out problem” means for us, exactly. Let’s assume that a one- or two-letter word (or word part) is placed by T<sub>E</sub>X at the end of a line. This word *sticks out* if the line without the word and its preceding interword space is longer than both neighboring lines and the line is recognizable as neither the last nor the next to last.

This definition declares a word part with two letters and a hyphen at its end as short. And the word doesn’t stick out if instead of an interword space a hyphen or a dash connects it to the previous word in the line. The last two lines are excluded but that doesn’t mean that a short word at their end looks good or is in any way acceptable if it sticks out.

For later use: Longer words that are captured in a rhythmical pattern can be tolerated. That is,

we look at a five-line block if the neighboring lines are shorter than the line with the floating word. And this scenario is acceptable if the first and/or last line of this block have at least nearly the same length as the middle line.

Let’s try to find out under which circumstances a short word isn’t moved to the next line in a normal text. To keep it simple let’s assume that neither penalties nor additional demerits are involved. Then the demerits for a line are computed by

$$(\backslash\text{linepenalty} + \backslash\text{badness})^2; \quad (1)$$

see [8, pp. 97–98]. With fixed-width interword spaces the badness of a line is represented by the used amount of the stretchability of the `\rightskip` [8, p. 30]. In other words: The badness can be read off from the length of a line. Different line lengths usually have different badness values if the stretchability of `\rightskip` isn’t extremely large: The longer a line the smaller the badness, except for the last line whose badness is always 0 if its width is less than `\hsize` and `\parfillskip` has its default setting.

So we have a line of badness  $\beta$  if the word that sticks out (let’s call it **w**) and the space in front of it are deleted. The second line is shorter than this line. Thus its badness is larger:  $\beta + \kappa$ . And the first line with **w** has a smaller badness:  $\beta - \mu$ . If **w** is moved to the second line the badness of this line gets smaller too:  $\beta + \kappa - \nu$ .

We can assume  $\beta > \mu > 0$  and we know  $\kappa \geq 0$  as the second line without **w** is shorter than the first line without **w** and that  $\mu \geq \nu - \kappa$  as the first line with **w** is longer than the second with **w**. And we have  $\nu \geq \mu$  as the badness is based on a cubic function (see [8, p. 97] or equation (3) in section 5) and a longer line reduces its badness by a smaller amount than a shorter if the same amount of text is added.

Let’s name the `\linepenalty`  $\lambda$ ; then (1) gives two sums for the demerits of the two lines. If **w** sticks out the demerits of the two lines are

$$(\lambda + \beta - \mu)^2 + (\lambda + \beta + \kappa)^2$$

and when **w** is moved to the second line we get

$$(\lambda + \beta)^2 + (\lambda + \beta + \kappa - \nu)^2.$$

The sticking-out problem is avoided if the second sum is smaller than the first.

$$\begin{aligned} (\lambda + \beta - \mu)^2 + (\lambda + \beta + \kappa)^2 &> (\lambda + \beta)^2 + (\lambda + \beta + \kappa - \nu)^2 \\ \iff \mu^2 - 2(\lambda + \beta)\mu &> \nu^2 - 2(\lambda + \beta + \kappa)\nu \\ \iff (\mu - 2\lambda - 2\beta)\mu &> (\nu - 2\lambda - 2\beta - 2\kappa)\nu. \end{aligned}$$

The negative value of the right-hand side does not get smaller if the factor  $\nu$  is replaced by  $\mu \leq \nu$ . Thus

$$\begin{aligned} \iff \mu &> \nu - 2\kappa \\ \iff \mu + \kappa &> \nu - \kappa \end{aligned}$$

which is always true if  $\kappa > 0$  as  $\mu \geq \nu - \kappa$ . In other words: A word cannot stick out if the following line has a larger badness than the line without the word.

But badness is computed by a heuristic. So a line might be shorter than another although both have the same badness value.

### Example 2: Description

Show that a word in the third-last line can stick out.

#### TeX input

```
\tenpoint\raggedright\noindent
Look: Never odd or even OR: neve ro ddo reveN, I
mean: Never odd or even OR: never odd or even.
It's a palindrome!
```

#### TeX output

```
Look: Never odd or even OR: neve ro ddo reveN, I
mean: Never odd or even OR: never odd or even.
It's a palindrome!
```

The width of the second line is 216.63643 pt; the first line up to and including the comma measures 216.77533 pt. Thus the “I” sticks out. This happens as the badness values are small and they give identical demerits independent of a line break in front of “I”. (In older versions of TeX with, for example, a different formula to compute the demerits this was probably not the case. See [10, p. 94].)

How to avoid this unwanted sticking-out problem without manual intervention? The only way I know for this case is to change the sequence of how TeX selects the breakpoints; see [14, pp. 372–373]. So either use `\looseness` or a three-line specification for `\parshape` without effect.

### Example 2 continued: TeX definitions

```
\parshape 3 Opt \hsize Opt \hsize Opt \hsize
```

#### TeX output

```
Look: Never odd or even OR: neve ro ddo reveN,
I mean: Never odd or even OR: never odd or even.
It's a palindrome!
```

On the other hand if the second line has badness 0 independent of the contents then the calculation to avoid the sticking-out problem requires that  $(\lambda + \beta - \mu)^2 + \lambda^2 > (\lambda + \beta)^2 + \lambda^2$  which is only possible for  $\mu > 2\lambda + 2\beta$ . This implies a negative  $\lambda$  as  $\beta > \mu > 0$ . Thus, if the last line of a paragraph in our ragged-right scenario has badness 0, a sticking-out look alike can easily occur in the second-last line as shown in example 1 that uses  $\lambda = 10$ .

Therefore, the `\parshape` technique doesn't remove the ‘a’ that sticks out in example 1. But it can be avoided by changing the value of `\linepenalty`.

### Example 3: Description

Avoid the sticking out of ‘a’ in a paragraph of example 1.

#### TeX input

```
\linepenalty=-154
```

### TeX output

Another kind of problem is shown in the next line: The last word sticks out as the other two lines are a little bit shorter; it looks similar to the first lines. □

But such a value of `\linepenalty` means that TeX likes lines with a high badness, i.e., short lines for ragged-right paragraphs. This generates not very good-looking output (see also [15, pp. 410–411]). As (1) excludes penalties and additional demerits, other ways to avoid this sticking-out scenario exist in this case as, for example, the parameter `\adjdemerits` plays a rôle. The assignment of a negative value to this parameter helps: It must be set to  $\leq -24296$  to get the output of example 3. It isn't always advantageous to avoid a sticking-out word in the penultimate line by setting parameters to extreme values.

And there are other effects: Even if the second line is a little bit longer and has therefore a smaller badness value than the first, the visual impression that a word sticks out might occur.

### Example 4: Description

Show that a word in the third-last line might appear to stick out.

#### TeX input

```
\ninepoint\raggedright\noindent
See this: Never odd or even OR neve ro ddo
reveN, I mean: Never odd or even. AND Never
odd or even. It's a palindrome!
```

#### TeX output

```
See this: Never odd or even OR neve ro ddo reveN, I
mean: Never odd or even. AND Never odd or even.
It's a palindrome!
```

Thus one consequence of the above is that an author who uses TeX's `\raggedright` has to check the right margin of the typeset text and to fix unwanted effects like the sticking-out problem. □

## 4 What is good ragged-right typesetting?

What are the recommendations of typographers for ragged-right texts? They do not seem to be universally established in a way that they can be written down as “rules”. Different people describing what they understand as *good typography* for the same language might list different sets of recommendations. Is it best to find the intersection of several sets—the universal recommendations accepted by several typographers—or should a consistent set of a widely recognized typographer be used?

I decided to follow one typographer, Friedrich Forssman. For the German language I use a text that discusses many details and gives several examples of good and bad ragged-right outputs; it is the book [4] by Forssman and R. de Jong. The book [16],

co-authored by Forssman, contains additional examples. Of course, the recommendations are tailored for German texts and German readers. Therefore I compare the recommendations with the statements in R. Bringhurst's book [3], a much shorter text which aims for brevity [3, p.9], to identify significant differences for English texts. An example text by R. Southall is taken from *TUGboat* [13].

**Ten recommendations.** Here is the list of recommendations that I extracted from [4]. These recommendations are not copied verbatim from the book. They are my interpretation of the information given in the text and the examples.

**R1 “zone”.** A *zone* at the right margin is defined that must be entered by a line before it can be broken. That is, except for the last lines of the paragraphs the lines cannot get arbitrarily short; at least they must reach the left boundary of the defined zone. Its width is determined by the hyphenation parameters (see R9 “hyphens”) and manual reworking. In a narrow column it should not be too wide or the margin looks frayed out. [4, p. 152, p. 158, p. 159]

**R2 “flutter”.** Ideally short and long lines should occur in turn; the back and forth of the line ends should form a rhythmical pattern. Notably, the right margin inside the zone should not look like an unsuccessful try to reach justification. [4, p. 124, p. 152]

**R3 “exceed”.** The zone must not end at `\hsize` if the line lengths are varying considerably. Since the right margin is not evident in this case, the length of the longest line is allowed to be wider than `\hsize`. In two-sided printing, the longest lines of the reverse side might show through from its straight left margin. [4, p. 152]

**R4 “no shape”.** The end of the lines should not form a contour or a shape, to avoid distracting the reader from the text. With restricted hyphenation (see R9 “hyphens”, case b) even four consecutive lines with line lengths that increase in small steps are objectionable if the measure is wide enough and the line lengths can vary properly. [4, p. 154, p. 158]

**R5 “end of line”.** A short word should not occur at the end of a line if the neighboring lines are shorter, i.e., the sticking-out problem should be avoided; especially one-letter words must be eliminated. Longer words at the end of a long line that stick out should be supported at least by words from two lines above or below, i.e., from one neighboring long line (see R2 “flutter”), if the zone is not narrow. [4, pp. 152–154, p. 155, p. 157]

**R6 “last line”.** If the paragraphs have no indentation then the last line of each paragraph must leave enough white space between its end and the

left boundary of the zone if the start of paragraphs isn't signaled in another way. That is, the end of a paragraph must be identifiable and it should not look like a line that has the minimal length. The distance from the left boundary of the zone should be between 2 and 4 em. [4, p. 143, p. 154]

**R7 “indent”.** In small columns an indentation of the first line is often unwelcome. [4, p. 143, p. 154]

**R8 “spaces”.** The interword spaces are all of equal width. Only their natural widths are used, the settings for their stretch- and their shrinkability are ignored. [4, p. 124, p. 146, p. 152, p. 154]

**R9 “hyphens”.** Hyphenation occurs in two forms. a) The settings of justified text for the length of the parts can be kept. Larger stacks of hyphens, i.e., consecutive lines ending with a hyphen or a dash, are tolerated than in justified texts.

b) Long syllables and *good* hyphenation points are preferred, i.e., the break shall respect the semantic structure of the word. In this case it is recommended to have at least four characters (and the hyphen) on the first line and at least four on the second; in this article, this structure is named (4, 4). Even better are five letters in both cases, i.e., (5, 5). That is, words that are broken must have at least eight or better ten letters. [4, p. 124, p. 155, p. 158]

**R10 “exceptions”.** In an emergency R8 “spaces” and R9 “hyphens” might be violated but only in a very limited way. Interword spaces might not be fixed: A very small amount, at most 0.015 em, can be used to *shrink* the interword spaces but only if no other solution is found. In R9b the pair (3, 4) instead of (4, 4) might be used. [4, p. 124, p. 155]

Measurements in the examples of [4] with line widths between 10.5 pc and 33 pc show that the zone widths vary from 23 pt to 57 pt; 48 pt for the widest measure. The ratio of zone width to `\hsize` lies between 12 and 20%; 27% creates a frayed out margin.

The two cases in R9 “hyphens” mark the major distinction why in German two words are in use for ragged-right typesetting as each case implies the settings for other recommendations. With case R9a the right margin becomes more compact and the line widths vary not much; this implements *Rauhsatz*. It is the method of choice if the `\hsize` is small. The case R9b is considered to be the *artistic* ragged-right typesetting, the *Flattersatz*. In this setup R2 “flutter” should be obeyed as strictly as possible.

What does “as strictly as possible” mean? The recommendations describe an *ideal* that an author cannot achieve perfectly in all cases. In two examples on pages 152 and 154 of [4] three to five lines appear in blocks of nearly the same line length, page 20

of [16] has five lines with increasing lengths, and page 91 contains a 5-line block in which the lengths of first three lines increase in small steps, the lengths of the fourth and fifth decrease by a small and a large width, respectively. So it builds a shape but isn't as eye-catching as the contour in example 1.

As indicated above, other typographers recommend more or other criteria. For example, [2, p. 45] adds to R2 “flutter” that at least one third of the lines shall be “sufficiently” filled.

**The comparison.** The book [3], p. 29, gives a self-involved example of ragged-right typesetting. It distinguishes between a ragged-right margin looking like a “neatly pinched piecrust” that is generated with the default typesetting parameters for hyphenation, spaces, etc., and a “hard rag” which is applied in the book. Let's check whether the ten recommendations above are covered.

R1 “zone”: no (and yes). A zone isn't mentioned and the text contains the statement that “no minimum line” is defined. But of course, the shortest non-last line and the `\hsize` form a zone.

R2 “flutter”: no. This topic isn't described in the text and the example doesn't obey it.

R3 “exceed”: unknown. Again not mentioned in the text but as the paragraph with the ragged-right margin appears between justified paragraphs no line exceeds the measure.

R4 “no shape”: unknown. The aspect isn't mentioned in the text. But the example has once four lines whose lengths increase in small steps. Thus the trend goes toward: no.

R5 “end of line”: unknown. There are no long lines in the sense of R2 “flutter” so the concept doesn't seem to be applicable. But the example has no short words that stick out at the end of a line.

R6 “last line”: probably yes. Again not explicitly mentioned in the text but the last line of the example is much shorter than the other lines.

R7 “indent”: unknown. Indentation is in general recommended on page 39 without mentioning ragged right. The example uses a wide measure and starts with an ornament to clearly distinguish it from the indented justified paragraphs.

R8 “spaces”: yes. The text states to use “fixed word spaces”.

R9b “hyphens”: yes. The text requires that hyphenation can occur only at explicit hyphens or at manually inserted hyphenation points.

R10 “exceptions”: unknown: not mentioned in the text and not obviously applied in the example.

Half of the recommendations cannot be decided because of lack of information; three times there

seems to be an agreement, and two have received a “no”. These two topics aren't explained in the book. That there is no minimum line has mainly to do with the unwillingness to hyphenate words but there is no short line without need.

In total I cannot say that the ideal of [3] for ragged-right typesetting is reflected by the ten recommendations that I extracted from [4]. The concept of alternate line length with its consequences isn't found in [3]. The text [13] doesn't help. Learning by example is in this case suboptimal: The first paragraph has alternating line lengths but two paragraphs later a contour is built; the last paragraph in section 5 has five strictly increasing line lengths and two paragraphs later a two-letter word sticks out.

**Recommendations vs. `\raggedright`.** Let's look at plain `TEX` first. The assignment `\rightskip=0pt plus 2em` implements R1 “zone”. This defines the zone width before the text is manually adjusted; in a strict sense it's violating R1. The zone measures 2 em in the first pass and  $\sqrt[3]{2} \times 2 \text{ em} \approx 2.52 \text{ em}$  in the second and third pass; see formula (2) below.

The other assignments `\spaceskip=0.3333em` and `\xspaceskip=0.5em` address R8 “spaces”, although, of course, two widths are used instead of a single interword space, this being for American English and not German.

Hyphenation is not affected by the macro, so it is the case R9a in which the settings of justified text are used; i.e., for German texts it implements `Rauhsatz`. All other recommendations are not covered.

As shown above, `LATEX2 $\epsilon$` 's `\raggedright` assigns infinite stretchability to `\rightskip`. This violates R1 “zone” because there is no controlled zone: All lines have badness 0 ([8, p.101]). `TEX`'s main criteria to judge about the quality of a line is lost. And as the interword spaces aren't fixed width the `LATEX` implementation violates R8 “spaces” too. To switch off indentation completely doesn't obey R7 “indent” which looks at the `\hsize`.

For better ragged-right typesetting — according to the ten recommendations — an author should use the (currently unmaintained) package `ragged2e`; see [12, pp.105–106]. It reassigns plain `TEX`'s value to `\rightskip` and provides new parameters to change not only this glue but also `\leftskip`, `\parindent`, and `\parfillskip`. And it sets fixed-width interword spaces so that it obeys R8 “spaces”.

**Out of scope.** The recommendations are for paragraphs but, of course, the page layout must be considered too. A page that contains paragraphs that are typeset ragged right with a wide zone needs a different optical alignment for a centered page number

at the bottom of the page than a page with justified text. On the other hand, in a two-column layout the white space between the columns (the gutter) might be smaller if the text is unjustified; it must only be larger than the interword space [4, p. 153]. A line that separates columns should be placed close to the straight margin [16, p. 93].

The opacity of the paper in double-sided printing has already been mentioned. Substantial transparency might let the straight left margin of the reverse page soothe the ragged-right margin [4, p. 152].

## 5 Implementation of the recommendations

As shown in the previous section plain  $\text{\TeX}$  provides a macro that puts its focus only on the zone and the interword spaces. The user has to decide how to handle aspects like indentation of the first line, hyphenation, etc., and thus the user must invoke other commands or change parameter values.

Before we decide what a plain  $\text{\TeX}$  command for ragged-right typesetting should do, techniques of  $\text{\TeX}$  are analyzed to consider how each recommendation can be implemented. In this analysis more than the obvious solution will be discussed.

Figure 1a) on the next page shows how plain  $\text{\TeX}$ 's `\raggedright` typesets the first paragraph of a fairy tale with a measure of 150 pt. The complete English text is available in [10, pp. 84–85, Fig. 6].

**R1 “zone”.** Above, an imperfect implementation of the zone was stated by using `\rightskip`. With a natural width of 0 pt and a finite stretchability the zone is defined up front. Even if the glue has to stretch more than its plain stretchability, i.e., the badness of the line is  $> 100$ , the `\tolerance` guarantees that there is a zone of limited width. Of course, this width must be chosen carefully with respect to the available line length. If `\pretolerance` equals `\tolerance`, the width of the zone does not depend on  $\text{\TeX}$ 's pass. If the stretchability of the `\rightskip` is named  $r^+$  and the current tolerance  $\tau$ , then the width of the zone in the pass is given by

$$\sqrt[3]{\frac{\tau}{100}} \times r^+. \quad (2)$$

With  $\tau = 100$ , only the specified amount  $r^+$  is available to fill the line. But we can make use of a little bit wider zone in a second pass. For example, if the width should be 20 pt in the first pass and 22 pt in the second pass, then we set `\pretolerance = 100` and `\tolerance = 133` as  $100(22/20)^3 = 133.1$  so that  $\sqrt[3]{1.33} \times 20 \text{ pt} \approx 1.0997 \times 20 \text{ pt} \approx 22 \text{ pt}$ .

And there are other ways to define a zone. One implementation is given in [7] ([10, pp. 93–94]). In

this setup `\rightskip` isn't used; instead the spaces are made active and become a sequence of three items: a horizontal skip with the stretchability for the zone and 0 pt natural width, a penalty of value 0, and a second horizontal skip, which has the natural width of an interword space and the negative stretchability of the first skip. It creates identical output to the version implemented in plain  $\text{\TeX}$  with `\raggedright`. (At the time [7] was written, `\rightskip` had not yet been invented [10, p. 154].)

A zone can also be defined using natural width and shrinkability of the same size for `\rightskip`. However, because glue cannot shrink more than the stated value, a `\tolerance` greater than 100 is without effect; thus there is only one zone width. Another disadvantage is that lines with less content have smaller badness values:  $\text{\TeX}$  prefers lines with lengths near the left boundary of the zone.

$\text{\TeX}$  prefers filled lines if the stretchability of the glue `\rightskip` defines the zone. The length of the line represents its badness: A line that reaches the left boundary of the zone has a higher badness than a line that needs all of `\hsize`. Is it good to treat them differently as both lines have made it into the zone? Let's assume that the tolerance is 0 so that all lines reaching the zone have equal badness.

If the heuristic to calculate the badness  $\beta$  for a line that stretches is interpreted as an equation

$$\beta = 100 \left( \frac{\text{used stretchability}}{\text{available stretchability}} \right)^3 \quad (3)$$

then we want to have  $\beta$  at most 0.49 so that it rounds down to 0. The available stretchability, that is, the stretchability of `\rightskip`, is then the maximal used stretchability, i.e., the zone width, divided by  $\sqrt[3]{0.49/100} \approx 0.17$ . Thus  $1/0.17 = 5.8875$  times the intended zone width has to be assigned to the stretchability of `\rightskip`. For example, a zone width of 20 pt needs a stretchability for `\rightskip` of 117.75 pt. It's hard to see any advantage compared to simply setting `\pretolerance = 100` and a direct assignment of the zone width to the stretchability of `\rightskip`. Several other parameters like penalties and additional demerits should be changed too but it isn't clear how to assign values to them to fix the situation. And if we think of the theory about the sticking-out problem then the scenario of example 2 becomes common.

**My implementation** defines the zone via the stretchability of `\rightskip` with a `\pretolerance` of 100 as it seems to be the best idea. Let's assume that the (temporary) dimen registers `\rrTd` and `\rrTdd` contain the width of the zone for the first pass and the width of the zone for the second



In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 1:** a) `\raggedright & \noindent`; b) and `\adjdemerits = -10000`; c) and `\adjdemerits = -40000`.

pass or 0pt, respectively. The second case is used if the default value 200 is used for the `\tolerance`.

```
\pretolerance=100 % define zone width of 1st pass
\advance\rightskip by 0pt plus \rrTd minus 0pt
\ifdim\rrTdd=0pt % use default \tolerance
  \tolerance=200
\else
  \ifdim\rrTdd>\rrTdd \errhelp=\rrEbadtolerance
  \errmessage{Width for 2nd pass must not be
    smaller than for 1st}%
  \tolerance=100
\else % set \tolerance for 2nd zone width
  \rrHcompbad\rrTdd=100(\rrTdd/\rrTdd)^3
  \rrHassign\rrTdd=[\rrTdd]
\fi\fi
```

The code uses two support macros directly and then two more are called. Two count registers `\rrTc` and `\rrTcc` appear in the code snippet.

```
\def\rrHassign#1=[#2]{% #1: count register
  % #1 gets integer part of (dimen #2/pt)
  \expandafter#1\expandafter
  \rrHinteger\the#2\end\ignorespaces}
\def\rrHcompbad#1=100(#2/#3)^3{%
  % #1: dimen register; #2, #3: dimens
  #1=#3\relax \rrHassign\rrTc=[#1]
  #1=#2\relax \rrHassign\rrTcc=[#1]
  % \rrTc=int(#3/pt) and \rrTcc=int(#2/pt)
  #1=\rrTcc pt % #1=int(#2/pt)pt, short: #1=#2
  \rrHmuldiv#1*\rrTcc/\rrTc\end % #1=#2^2/#3
  \rrHmuldiv#1*\rrTcc/\rrTc\end % #1=#2^3/#3^2
  \rrHmuldiv#1*100/\rrTc\end % #1=100*#2^3/#3^3
  \advance#1 by 0.5pt \ignorespaces}% round up
```

The next two support macros have simple implementations.

```
\def\rrHinteger#1.#2\end{#1\ignorespaces}
\def\rrHmuldiv#1*#2/#3\end{%
  % #1: dimen register; #2, #3: numbers
  \divide #1 by #3\relax
  \multiply #1 by #2\ignorespaces}
```

**R2 “flutter”.** If we want to have lines of different length then we want lines of different badness. A short line has a large badness, a long line gets a badness near 0. So we want very loose lines to be followed by decent lines and vice versa. The parameter `\adjdemerits` with value 10000 tries to guide  $\text{\TeX}$ 's line-breaking algorithm to avoid such combinations, that is, we need to change its value. Although the value of  $-10000$  makes  $\text{\TeX}$  work harder to find the best line breaks it seems to be worth assigning a negative value in order to support this recommendation. The paragraph in Fig. 1b) has one less hyphenated line end. Don't overreact: smaller values might give worse results for this aspect as Fig. 1c) shows.

An assignment to `\rightskip` for R1 “zone” is applied to all lines of a paragraph. There is no way to change it for odd and even line numbers. If the zone is implemented via active spaces something can be implemented to change the stretchability. Unfortunately, it is impossible to know where the line breaks occur, so whatever scheme is implemented for the switch it is guesswork.

Another possibility is to change `\hsize`. Well, this parameter is fixed for all lines of a paragraph like the `\rightskip`. But  $\text{\TeX}$  has the command `\parshape` which can be used to define independent line lengths. Of course, the maximal number of lines

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 2:** a) short is `\hspace{-6pt}` b) and start short; c) short is `\hspace{-9pt}`.

that the paragraphs of the text will reach must be set in advance to have different line lengths in every paragraph. Therefore a limit is declared that a user can increase to cover all number of lines in the paragraphs. As the `\parshape` command is reset after each paragraph [8, p. 103], its specification must be activated for each paragraph by `\everypar`.

But if we have a fixed parshape that, for example, indents the first line there is no chance to start a paragraph without indentation. Therefore the parshape specification must be built anew for each paragraph. Besides the indentation two other aspects should be made flexible. First, it should be possible to switch off the specification and to use `\hspace` for all lines. This helps to avoid the sticking-out problem as shown in example 2. Second, the sequence of alternate line lengths should be able to start with a short line instead of a long line. This might give more choices for the first line break.

Figure 2 shows three outputs with different line lengths for odd and even numbered lines. It uses plain TeX's `\raggedright`. The figure adds only in each case a parshape specification. Thus the observed zone width is increased by the amount that short lines are shorter than long lines.

**My implementation** consists of two parts. As mentioned above the parshape specification must be built anew for every paragraph because several aspects should be covered: (1) indent or not, (2) begin with a long or a short line, and (3) use only long lines, i.e., create a neutral `\parshape`.

Let `\rrCmax` be a constant that is larger than the number of lines of any paragraph in the text. We

need some parts to build the lines for `\parshape`: `\rrPl` is a single long line without indent, i.e., it is `Opt \hspace`, `\rrPil` is an indented long line, `\rrPs` is a short line without indent, and `\rrPis` is short and indented. Two blocks of line pairs, each of them with `\rrCmax/2-2` pairs, are built for long and short line specifications and for long lines only. The first block is named `\rrPpair`, the second `\rrPlong`.

The creation of the parshape parts is not complicated. Assume that `\rrTd` contains the amount by which the short lines are shorter than `\hspace` and `\rrTdd` contains the value for the indentation. `\rrTddd` is a third dimen register.

```
\rrTddd=\hspace           % create long lines
\edef\rrPl{Opt \the\rrTddd}%      no indent
\advance\rrTddd by -\rrTd
\edef\rrPil{\the\rrTdd\space\the\rrTddd}% indent
\rrTddd=\hspace \advance\rrTddd by -\rrTd
\edef\rrPs{Opt \the\rrTddd}% short, no indent
\advance\rrTddd by -\rrTd
\edef\rrPis{\the\rrTdd\space\the\rrTddd}% indent
```

The two blocks of long/short and long/long line pairs are created via a loop.

```
\edef\rrPpair{\rrPl\space\rrPs}% pair long/short
\edef\rrPlong{\rrPl\space\rrPl}% two long lines
\rrTc=\rrCmax \advance\rrTc by -4
\loop
\ifnum\rrTc>0 \advance\rrTc by -2
  \edef\rrPpair{\rrPpair\space
                \rrPl\space\rrPs}% add a pair
  \edef\rrPlong{\rrPlong\space
                \rrPl\space\rrPl}% here too
\repeat
```

The macro that builds the parshape specification makes use of three flags for the above mentioned cases: `\ifrrInoi` which is true if no indentation is wanted, `\ifrrIsl` which is true if the first line is short, and finally `\ifrrIlongonly` which is true if the alternate line lengths shall be ignored.

```
\def\rrMparshape{%
\begingroup          % needed for \aftergroup
\aftergroup\parshape % start \parshape
\aftergroup\rrCmax   % number of lines
\aftergroup\space
\ifrrIlongonly      % check for one length
\ifrrInoi           % check for indent
  \aftergroup\rrPl  % no indent
\else
  \aftergroup\rrPil% indent
\fi
\aftergroup\rrPl    % the other lines
\aftergroup\rrPlong
\else
\ifrrInoi           % check for indent
\ifrrIsl            % check start short
\else
  \aftergroup\rrPl
\fi
\aftergroup\rrPs % pair if started long
\else
\ifrrIsl            % indent short
  \aftergroup\rrPis
\else               % indent long, a pair
  \aftergroup\rrPil
  \aftergroup\rrPs
\fi\fi
\aftergroup\rrPpair% add a lot of lines
\ifrrIsl
  \aftergroup\rrPl % complete a short start
\fi\fi
\endgroup}
```

This macro is invoked for each paragraph. (A box from `\indent` is removed using `\lastbox`.)

```
\everypar={{\setbox0=\lastbox}\rrMparshape}%
```

**R3: extended zone.** The parameter `\hfuzz` might be used to allow  $\TeX$  to go beyond `\hsize` [8, p. 30]. The command does not increase the line length; it merely suppresses an overfull-line warning as long as the line sticks out less than the width given by `\hfuzz`. Thus this parameter is only used if  $\TeX$  isn't able to typeset the text without error; its value is not available for  $\TeX$ 's decisions and that is not exactly what is recommended. The dimen `\hfuzz` should be kept as an emergency tool for the user and not be used in a setup.

If the `\parshape` command is used then the length of the long lines can be changed independently from the line length of the shorter lines. But

then  $\TeX$  tries to fill all long lines beyond `\hsize` as this gives the lowest badness. Again this is not exactly what is recommended.

The third idea avoids these disadvantages: Add shrinkability to `\rightskip`. If a line uses its natural width or stretches then nothing is changed. But now a line can shrink too. The shrinkability  $r^-$  of the `\rightskip` adds to the total line length if the line shrinks. Long lines can therefore reach the length `\hsize + r^-`. Of course,  $r^-$  is added for long and short lines if a parshape specification is used.

Figure 3 on the next page applies  $r^- = 2\text{pt}$  and  $r^- = 4\text{pt}$  to previously shown examples.

**My implementation** changes, i.e., advances, `\rightskip` with shrinkability. If the dimen register `\rrTd` contains the amount by which lines shall be extended then the following assignment does the job:

```
\advance\rightskip by 0pt plus 0pt minus \rrTd
```

Such simple assignments to  $\TeX$  parameters are not shown anymore in the following subsections; the description in plain English should be sufficient.

**R4: no shape.** The first three recommendations are technical, thus a translation into  $\TeX$  macros and parameters was not too difficult, although a robust proof of their usefulness is lacking. However, the fourth recommendation is highly subjective.

Using long and short lines seems to avoid most shapes if the line lengths are sufficiently different. But, for example, the sequence of a short short, a short long, a long short, and a long long line might create the unwanted case of four lines with increasing length. More complicated designs for `\parshape` might avoid this effect. Or a negative `\adjdemerits` can trade the effect for words that stick out.

**R5: short words at the end of a line.** This situation cannot be avoided in all cases. It was shown that plain  $\TeX$ 's `\raggedright` doesn't like words that stick out. And if it happens, a neutral parshape specification might help; such a specification is provided in the implementation of R2 "flutter".

But with lines of different length the conditions are changed and short words might appear at the end of long lines. This sometimes gets fixed if the sequence of long/short pairs is changed and the paragraph is started with a short line. The above stated implementation of R2 "flutter" allows this.

Of course, one can tie short words to their successors. But that should only be done to fix bad breaks.

**R6: last line.** It is well known that the last line receives  $\TeX$ 's `\parfillskip`; by default it is `0pt plus 1fil` [8, p.30]. But that is not acceptable if

<p>In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.</p>	<p>In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.</p>	<p>In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.</p>
--	--	--

**Figure 3:** a) Fig. 1a) with  $r^- = 2\text{pt}$ ; b) Fig. 1b) with  $r^- = 4\text{pt}$ ; c) Fig. 2c) with  $r^- = 2\text{pt}$ .

the last line should be much shorter, in order to be easily distinguished from the left boundary of the zone if the start of a paragraph isn't signaled by indentation or other methods. The natural width of `\parfillskip` can be set to the zone width of the second pass plus the difference of the widths of long and short lines,  $w$ , and at least 2em:

$$\sqrt[3]{\frac{\text{\tolerance}}{100}} \times r^+ + w + 2\text{em}. \quad (4)$$

The recommendation states more precisely that the last line shall be 2 to 4em shorter than the other lines if the `\hspace` allows this. Instead of using only the natural width of `\parfillskip` it seems better to add 4em instead of 2em in (4) and to use a shrinkability of 2em. This assigns some demerits if the last line is only 2em shorter so `\TeX` more likely avoids such a short difference.

Of course, this setting might produce overfull lines. But this is a good indicator that something must be fixed by the author. And it is easy to enter `\parfillskip=0pt plus 1fil` at the end of a paragraph to restore the default behavior.

Another common problem is that a new line with only a word part is created. A large value for `\finalhyphdemerits` might fix such a situation.

Figures 4a) and 4b) on the next page apply the ideas. The first column uses `\raggedright` with `\parfillskip=65pt plus 1fil minus 20pt` and the next column adds `\finalhyphdemerits=50000`.

**R7: no indentation for small `\hspace`.** The plain `\TeX` macro keeps `\TeX`'s default behavior for indentation of the first line of a paragraph. To change that either the `\parindent` is set to 0pt or all of

the paragraphs are started with a `\noindent`. A parshape specification similar to R2 “flutter” sets the indentation for all paragraphs to a given parameter and provides with a Boolean flag a way to start a paragraph without indentation. So the author must make the decision either not to indent all paragraphs or to manually enter a command to suppress the indentation.

As the test paragraph is typeset without indentation, Fig. 4c) goes the other way and uses an indentation of 20pt with plain `\TeX`'s `\raggedright`.

**My implementation** invokes the above mentioned Boolean flag to avoid indentation:

```
\let\rrnoindent=\rrInoitrue
```

**R8: fixed interword space.** As mentioned above, interword spaces are usually built from `\fontdimen` parameters that `\TeX` reads from the font metric file of the font, while a nonzero `\spaceskip` overrides these font parameters as glue specification of interword spaces and a nonzero `\xspaceskip` defines the interword space if the `\spacefactor` is  $\geq 2000$ . Both parameters store a glue specification with three components for the natural width, the stretchability, and the shrinkability. But in ragged-right typesetting only the natural width shall be used.

Of course, the interword spaces should usually not deviate much from the font designer's setting documented in `\fontdimen2`. Note that a specification in the unit “em” involves `\fontdimen6`, the width of the font's quad [8, p. 60].

The recommendation requests a single width for interword spaces, that is, the value of `\xspaceskip` should stay at 0pt.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 4:** a) changed `\parfillskip`; b) and `\finalhyphendemerits`; c) Fig. 1a) without `\noindent`.

**R9: hyphenation.** In case a) normal hyphenation rules are applied, so we can concentrate on R9b). It seems that the code of  $\TeX$  must be changed to get “sense-conveying hyphenation” (see [6]), so this cannot be implemented by macros.

The assignment of 10000 to the two parameters `\hyphenpenalty` and `\exhyphenpenalty` switches off hyphenation. Smaller values define the cost of hyphens and thus their desirability. The default value for both parameters is 50 [8, pp. 96–97].

The first parameter sets the penalty that  $\TeX$  uses in the formula for the line demerits [8, p. 98] if the break occurs either at a hyphen that was inserted by  $\TeX$  or at a discretionary break in the text that the author has entered with `\-`. The second parameter stands for the penalty that a break receives if it occurs after a hyphen or a sequence of hyphens inside the text, i.e., hyphens entered by an author as `-`, `--`, or `---`. In order to be able to influence hyphenation the value for `\hyphenpenalty` must be less than 10000. Then an author can insert discretionary breaks to improve the output. Hyphenation can also be influenced by changing the values of the parameters `\doublehyphendemerits` and `\finalhyphendemerits` [8, p. 98].

If a word is given to  $\TeX$ 's hyphenation algorithm it first checks that it is a word of length  $\geq \text{\code\leftthyphenmin} + \text{\code\rightthyphenmin}$ . Such a word might be split into parts of at least `\leftthyphenmin` letters on the first line and `\rightthyphenmin` letters on the second. In case R9b) it is recommended to assign one of the values (4, 4), or (4, 5), or (5, 5) to the pair `(\leftthyphenmin, \rightthyphenmin)` so

that only words that have more than eight, nine, or ten letters get hyphenated; see [8, p. 454].

Figure 5 on the next page shows the effects of some of these parameters.

**R10: emergency.** The two methods that are suggested can be easily realized by  $\TeX$  macros. The shrinkability of interword spaces can be used to implement the emergency shrink. Two assignments to `\leftthyphenmin` and `\rightthyphenmin` change the length of word parts during hyphenation (see R9). And the default `\parfillskip` can be used.

Above, the dimension parameter `\hfuzz` was explained; it shall be left to the author for an emergency. And `\emergencystretch` is another option for problematic situations. As interword spaces cannot stretch it extends the width of the zone.

**My implementation** makes one change permanent; the other works only for a single paragraph. The macro `\rremergencyshrink` is named similar to the macro `\emergencystretch` and makes a permanent change. The help from a less restricted hyphenation is for a single paragraph only; an author might not know how to switch back to the settings that were active before.

Let's assume that the current settings of the parameters `\spaceskip` and `\xspaceskip` were captured in the macro `\rrMresetspaces`. The macro `\rremergencyshrink` is implemented in two simple steps. First, `\rrTd` gets the value by which the interword spaces shall shrink.

```
\def\rremergencyshrink{% shrink interword spaces
  \afterassignment\rrMshrinkspaces \rrTd}
```

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 5:** Fig. 1a) and a) `\hyphenpenalty = 10000`; b) `hyphenmins (4, 4)`; c) Fig. 2c) with `hyphenmins (5, 5)`.

The second step can now test `\rrTd`; a negative value is not allowed and it must not be larger than `0.015em`; `\rrHabsdim` makes the value positive and gives a warning if a negative value occurs. We have to be careful not to apply the macro more than once to `\spaceskip`. In the code, `\rrTs` is a skip register.

```
\def\rrMshrinkspaces{%
  \rrHabsdim\rrTd=[\rrTd] % absolute value
  \ifdim\rrTd>0.015em \errhelp\rrEbadshrink
    \errmessage{Interword spaces shouldn't
      shrink more than 0.015em}\rrTd=0.015em
\fi
\rrTs=0pt plus 0pt minus \rrTd
\rrMresetspaces % only one change is allowed
\ifdim\spaceskip>0pt
  \advance\spaceskip by \rrTs
  \ifdim\xspaceskip>0pt
    \advance\xspaceskip by \rrTs
\fi\fi}
```

Here is the second macro `\rrusehyphenmins`; it changes `\lefthyphenmin` and `\righthyphenmin`:

```
\def\rrusehyphenmins=(#1,#2){% change hyphenmins
% #1: lefthyphenmin; #2: righthyphenmin;
\rrMdefaultshyphenmins % reset to initial values
\ifnum#1<\lefthyphenmin
  \message{^^J\string\lefthyphenmin\space
    is not changed}%
\else \lefthyphenmin=#1\relax \fi
\ifnum#2<\righthyphenmin
  \message{^^J\string\righthyphenmin\space
    is not changed}%
\else \righthyphenmin=#2\relax \fi
\ignorespaces}
```

The code uses `\rrMdefaultshyphenmins`; a new macro that sets the values of `\lefthyphenmin` and `\righthyphenmin` back to the settings defined for the current language. In order to switch back to the values set by the ragged-right scheme a reset macro for the two “hyphenmins” is also coded. It is named `\rrMresetshyphenmins`.

**Addendum.** Several times we saw that parameters have to be reset or tested for each paragraph. Thus, as the last code snippet, the new implementation of `\par` is presented.

```
\def\par{\endgraf % end the paragraph
  \ifnum\prevgraf>\rrCmax % test for more lines
    \message{^^JRagged right needs more lines:
      \the\prevgraf^^J}\fi
  \rrMresetshyphenmins % reset parameters & flags
  \rrInoifalse \rrIslfalse \rrIlonlyfalse
  \rrMresetparfillskip % see section 6
  \rrMresetparskip}% see section 8
```

R2 “flutter” uses a constant `\rrCmax` for the maximum number of lines that a paragraph should have. At the end of each paragraph this is now checked and a violation is reported.

## 6 A generator for ragged-right macros

Except for R2 “flutter” and R1 “zone” — if the zone widths instead of the tolerance values shall be processed — the recommendations that can be implemented need only one or two assignments to  $\text{\TeX}$ 's parameters. That many parameters influence the typesetting of `\raggedright` is documented by the 15 different outputs shown in Figs. 1 to 5. So one can

combine the assignments into a single macro with a similar structure as `\raggedright`.

But for my experiments I find it much more convenient to have a kind of input mechanism that assigns default values to unused parameters. Then the switch to a new scheme resets all values and only the defined ones affect the new scheme. In total eight macros are called: (1) to set the zone for 1st and 2nd pass, (2) to create the macros used to build the `\parshape` command with indent and the amount by which short lines are reduced, (3) to add natural width and shrinkability to `\parfillskip`, (4) to make the interword spaces fixed width, (5) to assign the minimal length of word parts for hyphenation, (6) to set the two penalties and (7) the two additional demerits parameters that are involved in hyphenation, and (8) to support the typesetting by changing `\adjdemerits` and by declaring an overshoot. The main macro has the following structure:

```
\def\defraggedrightscheme #1: #2\end{%
% #1: name of the scheme; #2: parameter set
\let\rrMresetparskip=\relax % make names
\let\rrMresetparfillskip=\relax % known for
\let\rrMresetthyphenmins=\relax % \par
\expandafter\def\csname #1\endcsname{% scheme
\rightskip = Opt plus Opt minus Opt % init
% macros for the recommendations
\rrMsetzone #2 zone 1st,2nd:(Opt,Opt) \end
\rrMsetparshape #2 %
flutter indent,short:(Opt,Opt) \end
\rrMsetparfillskip #2 %
last nw,sh:(Opt,Opt) \end
\rrMsetspaces #2 %
space word,punct:(Opt,Opt) \end
\rrMsetlrmns #2 %
hyphenmins left,right:(0,0) \end
\rrMsetpenalties #2 %
hyphenpens ex,im:(0,0) \end
\rrMsetdemerits #2 %
hyphendems dbl,fin:(0,0) \end
\rrMsetsupport #2 %
aid exceed,adj:(Opt,0) \end
% change \par; see ‘‘Addendum’’ in section 5
\rrMenhancepar}}
```

The three macros used in the definition of `\par` (see the ‘‘Addendum’’ in section 5) get the meaning `\relax` so that `\par` can be executed. As unused parameters are reset to plain TeX’s defaults the call `\defraggedrightscheme justified:\end` creates `\justified` that ends TeX’s `\raggedright` and restores justified typesetting and

```
\defraggedrightscheme rrplain:
zone 1st,2nd:(2em,Opt)
space word,punct:(0.3333em,0.5em)\end
```

makes `\rrplain` equivalent to TeX’s `\raggedright`; only `\par` keeps its new coding. As a more complex example, here is the specification for Fig. 3c):

```
\defraggedrightscheme figIIIc:
zone 1st,2nd:(2em,Opt)
space word,punct:(0.3333em,0.5em)
flutter indent,short:(Opt,9pt)
aid exceed,adj:(2pt,10000)\end
```

It doesn’t seem to be necessary to show all eight macros. Larger parts of the code for the first two were presented above, so let’s concentrate on the third to explain the general idea.

```
\def\rrMsetparfillskip
#1last nw,sh:(#2,#3) #4\end{% def last line
% #1: some other parameters (not used here)
% #2: natural width of \parfillskip
% #3: its shrinkability
% #4: like #1 or empty: then use defaults
\def\next{#4}\ifx\next\empty
\parfillskip = Opt plus 1fil minus Opt
\else % use only positive dimen parameters
\rrHabsdim\rrTd=[#2] \rrHabsdim\rrTdd=[#3]
\parfillskip = \rrTd plus 1fil minus \rrTdd
\fi
\edef\rrMresetparfillskip{% to the new setting
\parfillskip=\the\parfillskip}}
\def\rrplainsparend{\parfillskip Opt plus 1fil
\par}% omit new specification in an emergency
```

If the `\parfillskip` isn’t set in a ragged-right scheme, i.e., there is no ‘‘last nw,sh:( $x, y$ )’’ in its definition, then #4 is empty as the keywords of the `\def` are matched by the strings given at the end of `\rrMsetparfillskip` in the code of the above listed macro `\defraggedrightscheme`. Otherwise this instance must occur in #4 and the statements in the `\else` are executed. So #2 and #3 define the natural width and the shrinkability of `\parfillskip`.

The macro `\rrMresetparfillskip` assigns the values of the scheme for `\parfillskip` with every `\par` (see ‘‘Addendum’’ in section 5).

## 7 Experiments

As one might expect, no parameter setting produces the ideal that the ten recommendations describe. Not only that some cannot be mapped into TeX macros, the developed macros cannot guarantee that their output avoids all shapes, words that stick out, etc. Moreover, other problems remain, for example, overfull lines might occur that must be fixed. Hence the author must accept compromises.

Thus our interest must be to reduce manual adjustments as much as possible. This section concentrates on the macros; only a few ties are used. Manual line breaks are discussed in section 8.

An attempt at ragged-right typesetting

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 6:** T<sub>E</sub>X's `\raggedright` and `\noindent`.

Okay, let's start with examples. *TUGboat*'s columns have a width of 225 pt. The `\raggedright` macro together with a `\noindent` produces for our single paragraph the output shown in Fig. 6. It is clear that too many hyphens occur for such a wide measure and too many of them appear in a sequence.

In a new ragged-right scheme for *TUGboat* I keep the width of the zone and the width of the interword spaces, including the larger space after certain punctuation marks. To make the lines flutter I introduce short lines which get the width `\hspace - 5pt`. As the measure is wide but indentation suppressed the last line should be much shorter than any short line. Finally, hyphenation shall only be applied to words with at least eight characters and even then it should cost more than in plain T<sub>E</sub>X. Moreover, consecutive hyphenated lines and a hyphenated second last line shall be very expensive. As the text is mixed with justified paragraphs I don't apply the `exceed` parameter.

Thus I use the following scheme for *TUGboat*:

```
\defraggedrightscheme secVII:
zone 1st,2nd:(2em,0pt)
space word,punct:(0.3333em,0.5em)
flutter indent,short:(0pt,0.5em)
last nw,sh:(6.5em,2em)
hyphenmins left,right:(4,4)
hyphenpens ex,im:(100,200)
hyphendems dbl,fin:(20000,50000)\end
```

Figure 7 shows the output if the test paragraph is typeset with this scheme: With the same number of lines all hyphens disappear and more lines approximate the right margin. But there is no proper alternation of line lengths.

Whenever I see this effect I first try to start with a short line if “flutter” is specified in the ragged-right scheme: Figure 8a) shows the result. The line lengths are varying a little bit better and more lines are sufficiently filled; but new hyphens occur. As

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 7:** Using the scheme `\secVII`.

specified, the word parts have at least four characters. The new hyphen in the second last line can only be avoided if a `\break` is inserted somewhere as it is the only way to typeset the paragraph with this scheme. A change to `\parfillskip` can change this with the risks that a lot of new line breaks occur and that the start of the next paragraph is not easily identified. Figure 8b) adds `\rrplainsparend` (see section 6) at the end of the paragraph. It lengthens the last line, changes all but five line breaks, and creates a single-letter word that sticks out.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

a) The scheme `\secVII` and `\rrstartshort`.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 8:** b) Like a) but with the default `\parfillskip`.



## Der Froschkönig

### oder der eiserne Heinrich

In den alten Zeiten, wo das Wünschen noch geholfen hat, lebte ein König, dessen Töchter waren alle schön, aber die jüngste war so schön, daß die Sonne selber, die doch so vieles gesehen hat, sich verwunderte sooft sie ihr ins Gesicht schien. Nahe bei dem Schlosse des Königs lag ein großer dunkler Wald, und in dem Walde unter einer alten Linde war ein Brunnen: wenn nun der Tag sehr heiß war, so ging das Königskind hinaus in den Wald und setzte sich an den Rand des kühlen Brunnens: und wenn sie Langeweile hatte, so nahm sie eine goldene Kugel, warf sie in die Höhe und fing sie wieder; und das war ihr liebstes Spielwerk.

Nun trug es sich einmal zu, daß die goldene Kugel der Königstochter nicht in ihr Händchen fiel, das sie in die Höhe gehalten hatte, sondern vorbei auf die Erde schlug und geradezu ins Wasser hineinrollte. Die Königstochter folgte ihr mit den Augen nach, aber die Kugel verschwand, und der Brunnen war tief, so tief, daß man keinen Grund sah. Da fing sie an zu weinen und weinte immer lauter und konnte sich gar nicht trösten. Und wie sie so klagte, rief ihr jemand zu ‘was hast du vor, Königstochter, du schreist ja daß sich ein Stein erbarmen möchte.’ Sie sah sich um, woher die Stimme käme, da erblickte sie einen Frosch, der

seinen dicken häßlichen Kopf aus dem Wasser streckte. ‘Ach, du bist, alter Wasserpatscher,’ sagte sie, ‘ich weine über meine goldene Kugel, die mir in den Brunnen hinabgefallen ist.’ ‘Sei still und weine nicht,’ antwortete der Frosch, ‘ich kann wohl Rat schaffen, aber was gibst du mir, wenn ich dein Spielwerk wieder heraufhole?’ ‘Was du haben willst, lieber Frosch,’ sagte sie, ‘meine Kleider, meine Perlen und Edelsteine, auch noch die goldene Krone, die ich trage.’ Der Frosch antwortete ‘deine Kleider, deine Perlen und Edelsteine, und deine goldene Krone, die mag ich nicht: aber wenn du mich lieb haben willst, und ich soll dein Geselle und Spielkamerad sein, an deinem Tischlein neben dir sitzen, von deinem goldenen Tellerlein essen, aus deinem Becherlein trinken, in deinem Bettlein schlafen: wenn du mir das versprichst, so will ich hinuntersteigen und dir die goldene Kugel wieder heraufholen.’ ‘Ach ja,’ sagte sie, ‘ich verspreche dir alles, was du willst, wenn du mir nur die Kugel wiederbringst.’ Sie dachte aber ‘was der einfältige Frosch schwätzt, der sitzt im Wasser bei seines Gleichen und quackt, und kann keines Menschen Geselle sein.’

Der Frosch, als er die Zusage erhalten hatte, tauchte seinen Kopf unter, sank hinab, und über ein Weilchen kam er wieder heraufgerudert; hatte die Kugel

im Maul und warf sie ins Gras. Die Königstochter war voll Freude, als sie ihr schönes Spielwerk wieder erblickte, hob es auf und sprang damit fort. ‘Warte, warte,’ rief der Frosch, ‘nimm mich mit, ich kann nicht so laufen wie du.’ Aber was half ihm, daß er ihr sein quak quak so laut nachschrie, als er konnte! Sie hörte nicht darauf, eilte nach Haus und hatte bald den armen Frosch vergessen, der wieder in seinen Brunnen hinabsteigen mußte.

Am andern Tage, als sie mit dem König und allen Hofleuten sich zur Tafel gesetzt hatte und von ihrem goldenen Tellerlein aß, da kam, plitsch plitsch, plitsch plitsch, etwas die Marmortreppe heraufgekrochen, und als es oben angelangt war, klopfte es an der Tür und rief ‘Königstochter, jüngste, mach mir auf.’ Sie lief und wollte sehen wer draußen wäre, als sie aber aufmachte, so saß der Frosch davor. Da warf sie die Tür hastig zu, setzte sich wieder an den Tisch, und war ihr ganz angst. Der König sah wohl, daß ihr das Herz gewaltig klopfte, und sprach ‘mein Kind, was fürchtest du dich, steht etwa ein Riese vor der Tür und will dich holen?’ ‘Ach nein,’ antwortete sie, ‘es ist kein Riese, sondern ein garstiger Frosch.’ ‘Was will der Frosch von dir?’ ‘Ach lieber Vater, als ich gestern im Wald bei dem Brunnen saß und spielte, da fiel meine goldene Kugel ins Wasser. Und weil ich

**Figure 9a:** First part of the Brothers Grimm fairy tale “The Frog King” in German (see, e.g., [5, pp. 39–43]).

I wrote in the introduction that I developed the macros to typeset German texts. So the next experiment switches the language. The column width is again narrow (148 pt; 30 digits create an overfull line) as this is a typical case to use ragged-right typesetting. Figure 9 shows the typeset output (over two pages).

There are a couple of problems if a German text is typeset with plain T<sub>E</sub>X. First, the umlauts (like ä)

must be entered with an accent and second the eszet (ß) is a control sequence; words with these letters cannot be hyphenated in plain T<sub>E</sub>X. Moreover, for hyphenation in general, the wrong patterns are loaded. I suppressed wrong hyphenations by declaring a few exceptions: zauderte, goldenes, goldenen, zweitenmal, and Spiel-kamerad. Three words with umlauts that occur frequently — Königskind (king’s child), Königstochter (king’s daughter), Königssohn

so weinte, hat sie der Frosch wieder heraufgeholt, und weil er es durchaus verlangte, so versprach ich ihm, er sollte mein Geselle werden, ich dachte aber nimmermehr, daß er aus seinem Wasser heraus könnte. Nun ist er draußen und will zu mir herein.' Indem klopfte es zum zweitenmal und rief

'Königstochter, jüngste, mach mir auf, weißt du nicht, was gestern du zu mir gesagt bei dem kühlen Brunnenwasser? Königstochter, jüngste, mach mir auf.'

Da sagte der König 'was du versprochen hast, das mußt du auch halten; geh nur und mach ihm auf.' Sie ging und öffnete die Türe, da hüpfte der Frosch herein, ihr immer auf dem Fuße nach, bis zu ihrem Stuhl. Da saß er und rief 'heb mich herauf zu dir.' Sie zauderte bis es endlich der König befahl. Als der Frosch erst auf dem Stuhl war, wollte er auf den Tisch, und als er da saß, sprach er 'nun schieb mir dein goldenes Tellerlein näher, damit wir zusammen essen.' Das tat sie zwar, aber man sah wohl, daß sies nicht gerne tat. Der Frosch ließ sich gut schmecken, aber ihr blieb fast jedes Bißlein im Halse. Endlich sprach er 'ich habe mich satt gegessen und bin müde, nun trag mich in dein Kämmerlein und mach dein seiden Bettlein zurecht, da wollen wir uns schlafen legen.' Die Königstochter fing

an zu weinen und fürchtete sich vor dem kalten Frosch, den sie nicht anzurühren getraute, und der nun in ihrem schönen reinen Bettlein schlafen sollte. Der König aber ward zornig und sprach 'wer dir geholfen hat, als du in der Not warst, den sollst du hernach nicht verachten.' Da packte sie ihn mit zwei Fingern, trug ihn hinauf und setzte ihn in eine Ecke. Als sie aber im Bette lag, kam er gekrochen und sprach 'ich bin müde, ich will schlafen so gut wie du: heb mich herauf, oder ich sags deinem Vater.' Da ward sie erst bitterböse, holte ihn herauf und warf ihn aus allen Kräften wider die Wand, 'nun wirst du Ruhe haben, du garstiger Frosch.'

Als er aber herabfiel, war er kein Frosch, sondern ein Königssohn mit schönen freundlichen Augen. Der war nun nach ihres Vaters Willen ihr lieber Geselle und Gemahl. Da erzählte er ihr, er wäre von einer bösen Hexe verwünscht worden, und niemand hätte ihn aus dem Brunnen erlösen können, als sie allein, und morgen wollten sie zusammen in sein Reich gehen. Dann schliefen sie ein, und am andern Morgen, als die Sonne sie aufweckte, kam ein Wagen herangefahren mit acht weißen Pferden bespannt, die hatten weiße Straußfedern auf dem Kopf und gingen in goldenen Ketten, und hinten stand der Diener des jungen Königs, das war der treue Heinrich. Der treue Heinrich hatte

sich so betrübt, als sein Herr war in einen Frosch verwandelt worden, daß er drei eiserne Bande hatte um sein Herz legen lassen, damit es ihm nicht vor Weh und Traurigkeit zerspränge. Der Wagen aber sollte den jungen König in sein Reich abholen; der treue Heinrich hob beide hinein, stellte sich wieder hinten auf und war voller Freude über die Erlösung. Und als sie ein Stück Wegs gefahren waren, hörte der Königssohn, daß es hinter ihm krachte, als wäre etwas zerbrochen. Da drehte er sich um und rief

'Heinrich, der Wagen bricht.'  
'Nein, Herr, der Wagen nicht, es ist ein Band von meinem Herzen, das da lag in großen Schmerzen, als ihr in dem Brunnen saßt, als ihr eine Fretsche (Frosch) wast (wart).'

Noch einmal und noch einmal krachte es auf dem Weg, und der Königssohn meinte immer der Wagen bräche, und es waren doch nur die Bande, die vom Herzen des treuen Heinrich absprangen, weil sein Herr erlöst und glücklich war.

The following scheme was used:

```
\defraggedrightscheme flutterXXX:
zone 1st,2nd:(2em,3em)
flutter indent,short:(0pt,0.5em)
last nw,sh:(6.5em,2em)
space word,punct:(0.33333em,0pt)
hyphenmins left,right:(4,4)
hyphenpens ex,im:(50,500)
hyphendems dbl,fin:(10000,20000)
aid exceed,adj:(2pt,0)\end
```

**Figure 9b:** Second part of “The Frog King”, and the ragged-right scheme used to typeset the text.

(king’s son) — are entered with a discretionary hyphen, for example, `K\onigs\tochter`. This is necessary to avoid overfull lines. I made hyphenation expensive and both “hyphenmins” are set to 4; see recommendation R9. But nevertheless, the result cannot be optimal as a second pass isn’t able to utilize all possible hyphenation points.

For a German text I set all interword spaces to the same width, that is, the zero skip is used

for `\xspaceskip`. The gutter is set to only 11 pt. I allow lines to overflow by 2 pt so that at least 9 pt separate the columns. This is more than two times the interword space and should be sufficient for a reader to identify the end of a line. As the column width is small I decided to make the last line in the worst case only 1 em shorter than the left side of the zone in a short line. Short lines are 5 pt shorter than long lines. Thus the zone width measures  $\approx 35$  pt

The communications of the T<sub>E</sub>X Users Group are published irregularly at Providence, Rhode Island, and are distributed as a benefit of membership both to individual and institutional members.

Submissions to *TUGBOAT* are for the most part reproduced with minimal editing, and any questions regarding content or accuracy should be directed to the authors, with an information copy to the Editor.

The deadline for submitting items for Vol. 7, No. 2, is April 28, 1986; the issue will be mailed in late June.

Manuscripts should be submitted to a member of the TUGBOAT Editorial Committee. Articles of general interest, those not covered by any of the editorial departments listed, and all items submitted on magnetic tape or as camera-ready copy should be addressed to the Editor, Barbara Beeton.

Contributions in camera copy form are encouraged, as is electronic submission of items on magnetic tape, via electronic mail, or transferred directly to the AMS computer; for instructions, write or call Barbara Beeton.

I carefully read the November issue of *TUGBOAT* and was shocked to see that I did not acknowledge the contributions of Pierre MacKay, Michael Spivak, and the founders of our organization. I correct this by thanking them and the AMS for creation of a good productive organization. Organizations like TUG always depend upon some good people contributing of their time. These people have really set a solid foundation for us to build upon.

Michael Spivak has resigned from the Steering Committee since he is now strictly in the T<sub>E</sub>X community for financial gain. I am sure that he will continue to be available as a sounding board.

**Figure 10:** *TUGboat* 7:1 (1986) a) & b) five paragraphs from page 2; c) two paragraphs from page 7.

(only two lines are tight) or a rather large  $\approx 23\%$  of the `\hspace` to compensate for missed hyphens. See the end of Fig. 9 for the complete definition.

I made three additional changes. The text after the quotation in the first column of Fig. 9b was forced to start with a short line; same for the following paragraph. Moreover I added a tie to remove a short word that sticks out in the middle column of Fig. 9a: `verspricht, so~will`.

Of course, the result is not perfect; the ideal described by the above-discussed recommendations is not reached. But the output is much better than the result with T<sub>E</sub>X's `\raggedright` and `\noindent`. That creates four overfull lines, the start of two paragraphs are not identifiable as the last lines of the previous paragraphs are nearly filled, and many more lines end with a hyphen.

Back to English: *TUGboat* Volume 7:1, issue no. 14, published in March 1986, is special; see [1]. First, it has two guest editors, David Kellermann and Barry Smith, and second, a unique layout designed by Martha Gannett is used. On pages with three columns the column width seems to be 150 pt, the main font is `cmr9`, and all paragraphs are set ragged right. I assume the editors adjusted some line breaks manually. This offers an opportunity to see how the macros of this article handle this situation.

I took seven paragraphs from the issue: The first five are from page 2, the inside cover. These are the first five paragraphs in this issue. Two are copied from page 7, the first two paragraphs of the regular column "From the President". I don't know which macros and parameter settings were used by the ed-

itors. I guessed some values from the output and added others freely to try to achieve a good result, according to the recommendations we have seen.

```
\defraggedrightscheme issueXIV: % used with cmr9
zone 1st,2nd:(20pt,26pt) % guess
flutter indent,short:(20pt,5pt)
space word,punct:(3.3333pt,4.4444pt) % guess
hyphenmins left,right:(4,4)
hyphenpens ex,im:(50,500)
hyphendems dbl,fin:(10000,20000)
aid exceed,adj:(2pt,0)\end
```

Figure 10a) shows the first two paragraphs. The first has the same line breaks as the one published; the second is one line longer because line 4 has in the issue a width of more than 150.9 pt, i.e., it is overfull and only the high `\hfuzz` of 1 pt avoids the overfull bar. In Fig. 10a) this line is short and can only have 147 pt.

The three paragraphs in Fig. 10b) have identical right margins to the ones in the issue; the last one is started with a short line though. I typed `submit\~ted` as this hyphen isn't found with the (4,4) setting for the "hyphenmins". I accepted this as the hyphen occurs in the original too.

The first paragraph in Fig. 10c) is different from the original. It has the same length in lines but avoids a word that sticks out. A tie in this paragraph between "build upon" lengthens the last line; otherwise "set" in the penultimate line appears at the end of the third last line. The second paragraph would add the word "be" in the penultimate line compared to the output in issue 14. This was changed by using a tie between "be available".

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

**Figure 11:** Using a) `\secVII`;

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

b) `\secVIIIn`;

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

c) `\secVIIIn` and high penalties.

Let's look again at our single paragraph. Using the `\hsize` of Figs. 1–5 the scheme `\secVII` creates the output of Fig. 11a). The scheme `\secVIIIn` — that is similar to the `\secVII` but without a change to the penalty or demerits parameters for hyphens and with some overshoot — gives a very different result, as shown in Fig. 11b): The lines are filled with more material but four hyphens appear now. If the two penalties are set in the scheme `\secVIIIn` to the values of scheme `\secVII` a third set of line breaks is found; see Fig. 11c). Note the outputs in Fig. 11 differ from the 15 variants shown in Figs. 1–5.

```
\defraggedrightscheme secVIIIn:
zone 1st,2nd:(2em,0pt)
space word,punct:(0.3333em,0.5em)
flutter indent,short:(0pt,0.5em)
last nw,sh:(6.5em,2em)
hyphenmins left,right:(4,4)
aid exceed,adj:(2pt,10000)\end
```

## 8 Manual adjustments

There are at least two insights that one can learn from Fig. 11. First, each parameter influences the output; this is of course a well-known fact. Second, it seems to be impossible to find a set of parameters that works for a couple of paragraphs as each paragraph seems to be so sensitive to the chosen parameters of the scheme. That is, each paragraph might require its own set of parameters or a lot of manual intervention to be typeset in the best way.

Some commands to change the parameters temporarily have been presented in R10 “exceptions”.

The implementation of R2 “flutter” adds macros like `\rrstartshort` and others. Moreover, the usefulness of ties was already shown. Although an author possesses a lot of techniques for fine tuning it is necessary to add control sequences to our toolbox to allow manually inserted line breaks too.

The simplest forms are to enter `\hfil\break` or `\break`. But this often works only if an existing break is used; otherwise lines might become over- or underfull or previous line breaks change. It is better to use a variant of the answer to exercise 14.15 of *The T<sub>E</sub>Xbook* [8]. A command first ends the paragraph with a larger zone and with more overshoot. Then a new paragraph is started. This permits using a different parshape for the next lines. For example, the next line can be made long or short or the parshape could use long lines only.

```
\def\rrHbreakhere{%
  \parfillskip 0pt plus 1.5em minus 1.5em\par
  \parskip=0pt \rrnoindent}
\let\rrbreakls=\rrHbreakhere
\def\rrbreaksl{\rrHbreakhere\rrstartshort}
\def\rrbreakll{\rrHbreakhere\rrnoparshape}
\edef\rrMresetparskip{\parskip=\the\parskip}
```

The values of `\parfillskip` and `\parskip` are reset by `\par` so no group is needed. (See “Addendum” at the end of section 5.) The shrinkability of `\parfillskip` glue allows extending the line beyond the value of `\hsize` similar to the way `\rightskip` was changed in R3 “extended zone” of section 5. Of course, since these new control sequences are for fine control, the complexity to find the best action in problematic situations increases.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

The deadline for submitting items for Vol. 7, No. 2, is April 28, 1986; the issue will be mailed in late June.

Manuscripts should be submitted to a member of the TUGBOAT Editorial Committee. Articles of general interest, those not covered by any of the editorial departments listed, and all items submitted on magnetic tape or as camera-ready copy should be addressed to the Editor, Barbara Beeton.

Contributions in camera copy form are encouraged, as is electronic submission of items on magnetic tape, via electronic mail, or transferred directly to the AMS computer; for instructions, write or call Barbara Beeton.

**Figure 12:** a) one fix in Fig. 11a); b) four fixes in Fig. 11c); c) Fig. 10b) and two fixes.

Why 1.5 em in `\rrbreakhere`? Well, it is clear that the stretch- and shrinkability should depend on the font, thus the unit “em”. What is needed is to adjust the line breaks by moving a few characters from one line to another. The specification `\parfillskip 0pt plus 1fil minus \maxdimen` or something similar acts much more strongly but it might change earlier line breaks; for example, see Fig. 8b. This should not happen.

**Applications.** Does the sample paragraph require manual adjustment? If the recommendations of [3] are followed then the text should have no hyphens. The text was typeset without hyphens only four times in Figs. 1–5 and 11: 5a), 5c), 11a), and 11c). The simple case 5a) already fulfills all the explicitly stated criteria. Nevertheless, the longest line width in this output measures only  $\approx 145.6$  pt and not many lines reach a similar length. Fig. 11a) has the problem that a two-letter word sticks out. It can be fixed by applying `\rrbreaks1` after “shone”; the result is shown in Fig. 12a). Its longest line measures  $\approx 147.4$  pt. But most lines are much shorter.

And Fig. 5a) is far from obeying the recommendations for Flattersatz. The lines do not alternate well in their lengths and especially the first four lines build a “staircase down” violating R4 “no shape”. This problem occurs in Fig. 12a) too, although it does not have the second staircase in lines 13–16 anymore.

Applying the latest used scheme of Fig. 11c) I entered four line breaks: `\rrbreaks1` after “still” and `\rrbreak1s` after “was”, “took”, and “on”. And in the last line I suppressed the overfull line warning

with `\rrbreak11`. Thus, I tolerate a hyphen in the second line as it seems to be impossible to avoid all hyphens in the text and I accept that the start of the next paragraph might be difficult to identify.

It's simple to move the word “should” one line up in Fig. 10a) to make this paragraph identical to the one in the issue. A fix for the B-like contour that two paragraphs built in Fig. 10b) is more complex. With `\rrbreak1s` in each paragraph (after “submitted” and “copy”) the result of Fig. 12c) is accomplished. It has one more hyphenated line though.

## 9 Summary

To typeset a text with a straight-left and a ragged-right margin, typographers recommend certain settings for the interword spaces, the indentation and the length of the last line of a paragraph. Moreover, topics like hyphenation and the length of line pairs play a major rôle. The occurrence of a structure at the right margin adds a new aesthetic element but it should not attract the attention of the reader. Thus, it increases the complexity of finding a pleasing set of line breaks. And it raises the question: When does the ragged-right margin look acceptable?

There is more than one idea of how ragged-right output should look. German typographers use different words to distinguish two concepts. One works with small measures and accepts a lot of hyphenated words; this is called “Rauhsatz”. The other requests less hyphenation and can be used with larger `\hsizes`, even as a replacement for justified typesetting, the “Flattersatz”. English tradition for good ragged-right typesetting agrees with some aspects of

the Flattersatz but it seems to deviate not only for hyphenation but also in how to rate a ragged-right margin as good.

This article extracts ten recommendations for ragged-right typesetting from the statements of a well-known German typographer. The plain  $\text{\TeX}$  macro `\raggedright` implements only two of them. The article tries to identify for the others which parameter settings and macros support plain  $\text{\TeX}$ 's `\raggedright` to typeset text in Flattersatz.

1. Two assignments, one to the stretchability of `\rightskip` and one to the `\tolerance`, implement a zone once for the first and once for the second and third pass.
2. The creation of a `\parshape` command from pairs of lines with different line lengths seems to avoid most unwanted shapes of the right margin. But even a neutral `\parshape` helps to avoid words that stick out and is quite useful.
3. As no straight right margin is visible an assignment to the shrinkability of `\rightskip` lets lines extend beyond the `\hsize`.
4. If the start of a paragraph is mainly signaled by the length of the last line of the previous paragraph then the natural width and the shrinkability of `\parfillskip` should make last lines 2 to 4em shorter than any other line.
5. With narrow line lengths indentation should be avoided, i.e., `\parindent` set to 0pt if the indentation isn't specified in the `\parshape`.
6. Either the natural width of `\spaceskip` is used to get fixed-width interword spaces or the three `\fontdimen` 3, 4, 7 of all fonts are set to 0pt.
7. To control hyphenation, six parameters shall be considered: a) two penalties: `\hyphenpenalty` and `\exhyphenpenalty`, b) the length of word parts: `\lefthyphenmin` and `\righthyphenmin`, and c) two demerits: `\doublehyphendemerits` and `\finalhyphendemerits`. At least the values for the word parts should be larger than for justified typesetting; e.g., set both to 4.
8. In an emergency a very small amount of shrinkability can be used for `\spaceskip`:  $\leq 0.015$  em.
9. Use a `\parfillskip` that has small amounts of stretch- & shrinkability for manual line breaks.

An author must be aware that a lot of time-consuming manual adjustments to the text — applying among other things no. 9 — remain necessary.

## References

- [1] Barbara Beeton (ed.), *TUGboat* 7:1 (1986); guest editors David Kellermann and Barry Smith; layout design Martha Gannett.  
[tug.org/TUGboat/tb07-1/tb14complete.pdf](http://tug.org/TUGboat/tb07-1/tb14complete.pdf)
- [2] Max Bollwage, *Typografie kompakt*, Berlin, Germany: Springer-Verlag, 2001.
- [3] Robert Bringhurst, *The Elements of Typographic Style*, Seattle, Washington: Hartley & Marks Publishers, 4th edition, version 4.2, 2016.
- [4] Friedrich Forssman & Ralf de Jong, *Detailtypographie*, 3rd edition, Mainz, Germany: Verlag Hermann Schmidt, 2004.
- [5] Brüder Grimm, *Kinder- und Hausmärchen*, 14th edition, Darmstadt: Wissenschaftliche Buchgesellschaft, 1991; under license by Winkler Verlag, München, 1949.
- [6] Yannis Haralambous: “New hyphenation techniques in  $\Omega_2$ ”, *TUGboat* 27:1 (2006), 98–103.  
[tug.org/TUGboat/tb27-1/tb86haralambous-hyph.pdf](http://tug.org/TUGboat/tb27-1/tb86haralambous-hyph.pdf)
- [7] Donald E. Knuth and Michael F. Plass, “Breaking paragraphs into lines”, *Software — Practice and Experience* 11 (1981), 1119–1184; reprinted with an addendum as Chapter 3 in [10], 67–155.
- [8] Donald E. Knuth, *The  $\text{\TeX}$ book*, Volume A of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1984.
- [9] Donald E. Knuth, *The METAFONTbook*, Volume C of *Computers & Typesetting*, Boston, Massachusetts: Addison-Wesley, 1986.
- [10] Donald E. Knuth, *Digital Typography*, Stanford, California: Center for the Study of Language and Information, CSLI Lecture Notes No. 78, 1999.
- [11] Leslie Lamport, *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System*, 2nd edition, Reading, Massachusetts: Addison-Wesley, 1994.
- [12] Frank Mittelbach and Michael Goossens, *The L<sup>A</sup>T<sub>E</sub>X Companion*, 2nd edition, Boston, Massachusetts: Addison-Wesley, 2004.
- [13] Richard Southall, “First principles of typographic design for document production”, *TUGboat* 5:2 (1984), 79–90. Errata in *TUGboat* 6:1 (1985), 6.  
[tug.org/TUGboat/tb05-2/tb10south.pdf](http://tug.org/TUGboat/tb05-2/tb10south.pdf)  
[tug.org/TUGboat/tb06-1/tb11gendel.pdf](http://tug.org/TUGboat/tb06-1/tb11gendel.pdf)
- [14] Udo Wermuth, “Tracing paragraphs”, *TUGboat* 37:3 (2016), 358–373; Errata in *TUGboat* 38:3 (2017), 414 (see [15]).  
[tug.org/TUGboat/tb37-3/tb117wermuth.pdf](http://tug.org/TUGboat/tb37-3/tb117wermuth.pdf)
- [15] Udo Wermuth, “A note on `\linepenalty`”, *TUGboat* 38:3 (2017), 400–414; Errata in *TUGboat* 39:1 (2018), 87.  
[tug.org/TUGboat/tb38-3/tb120wermuth.pdf](http://tug.org/TUGboat/tb38-3/tb120wermuth.pdf)  
[tug.org/TUGboat/tb39-1/tb121wermuth-adem.pdf](http://tug.org/TUGboat/tb39-1/tb121wermuth-adem.pdf)
- [16] Hans Peter Willberg, Friedrich Forssman, *Lesetypographie*, 2nd printing, Mainz, Germany: Verlag Hermann Schmidt, 2005.

◇ Udo Wermuth  
Dietzenbach, Germany  
u dot wermuth (at) icloud dot com